



**Universitat Autònoma
de Barcelona**

SEGUIMENT D'OBJECTES AMB UN SISTEMA DE VISIÓ ARTIFICIAL

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes

realitzat per

David Perálvarez Fernández

i dirigit per

Ramon Baldrich Caselles

Escola d'Enginyeria

Sabadell, Juny de 2010

El sotasignat, Ramón Baldrich Caselles,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en David Perálvarez Fernández
I per a que consti firma la present.
Sabadell, Juny de 2010

Signat: Ramón Baldrich Caselles

Resum

Avui dia l'ésser humà viu immers en un món on la informàtica juga un paper importantíssim. Dintre de la informàtica hi ha un subcamp anomenat Visió per Computador.

No tothom sap de què tracta aquest subcamp, tot i així, hom ha tingut contacte directe algun cop amb ell. Hi ha molts casos que expliquen això:

Per exemple, quan s'entra a un pàrquing i aquest és capaç de guardar la matrícula mitjançant una càmera. Seria lògic pensar que la matrícula ha sigut recollida per una persona que està mirant la càmera en el moment que arriba el cotxe. Malgrat això, la realitat és ben diferent. La matrícula ha sigut detectada per un computador, ell solet ha sigut capaç de dir què hi apareix en ella i ho ha aconseguit gràcies a la utilització de tècniques de Visió per Computador.

Altres casos poden ser: la inspecció de peces en la cinta transportadora d'una indústria, on automàticament es detecta si la peça té algun defecte. O les càmeres digitals que ens fan la foto en el moment en què estem somrient.

Amb aquests exemples tan diferents es vol fer entendre, que la Visió per Computador abasta moltes aplicacions útils per les persones i que en un futur pròxim tindrà un gran impacte.

Altrament, aquest projecte s'ha centrat en tractar la part d'aquesta ciència que s'encarrega del seguiment d'objectes dins d'una seqüència d'imatges. Concretament, s'ha fet una aproximació al seguiment d'objectes dins d'una escena qualsevol.

Taula de continguts

Resum	1
 CAPÍTOL 1	 7
1.1. Motivacions	8
1.2. Objectiu	8
1.3. Organització de la memòria	9
 CAPÍTOL 2	 11
2.1. Introducció.....	12
2.1.1. Tipologia i paraules clau	12
2.1.2. Descripció	12
2.1.3. Objectius del projecte	12
2.1.4. Definicions, acrònims i abreviacions.....	13
2.1.5. Parts interessades	14
2.1.6. Producte i documentació del projecte	15
2.2. Estudi de la situació actual.....	15
2.2.1. Context	15
2.2.2. Normatives i legislació	16
2.3. Requisits del projecte.....	16
2.3.1. Requisits funcionals	16
2.3.2. Requisits no funcionals	16
2.3.3. Restriccions del sistema	17
2.3.4. Catalogació i priorització dels requisits	17
2.4. Alternatives i selecció de la solució.....	18
2.4.1. Alternativa 1	18
2.4.2. Alternativa 2	18
2.4.3. Alternativa 3	19

2.4.4. Solució proposada	19
2.5. Planificació del projecte	20
2.5.1. Recursos del projecte	20
2.5.2. Tasques del projecte	21
2.5.3. Planificació temporal inicial.....	23
2.6. Avaluació de riscos	24
2.6.1. Llista de riscos	24
2.6.2. Catalogació de riscos	24
2.6.3. Pla de contingència.....	25
2.7. Pressupost	26
2.7.1. Estimació cost de personal.....	26
2.7.2. Estimació cost dels recursos.....	26
2.7.3. Resum i anàlisi cost benefici	27
2.8. Conclusions de l'estudi de viabilitat	27
 CAPÍTOL 3	 29
3.1. Què és la Visió per Computador?	30
3.1.1. Introducció	30
3.1.2. Interpretació de la imatge.....	30
3.1.3. Tipus d'imatges	32
3.1.3.1. Monocromàtiques	32
3.1.3.2. Multiespectrals	32
3.1.4. Etapes d'un sistema de Visió Artificial	33
3.1.5. Aplicacions de la Visió Artificial	34
3.2. Imatges binàries.....	34
3.2.1. Introducció	34
3.2.2. Definició.....	35
3.2.3. Veïnatge	35
3.2.4. Etiquetatge	36
3.2.5. Morfologia binària	37

3.3. Filtre de Kalman	38
3.3.1. Introducció	38
3.3.2. Descripció	38
3.3.3. Formulació del filtre de Kalman Discret.....	39
3.3.4. Ús del filtre en la Visió per Computador	40
3.3.5. Algorisme	41
3.4. Sostracció del fons de la imatge	43

CAPÍTOL 4

4.1. Introducció.....	48
4.2. Descripció del problema del seguiment d'objectes	48
4.2.1. Model dinàmic.....	48
4.2.2. Comportament dels objectes	49
4.2.3. Evolució dels objectes	50
4.3. Paradigma de desenvolupament	52
4.4. Especificació de requeriments	54
4.4.1. Requeriments funcionals.....	54
4.4.2. Requeriments no funcionals	55
4.4.2.1. Requeriments de rendiment	55
4.4.2.2. Restriccions de disseny	56
4.4.2.3. Interfícies externes	56
4.5. Disseny modular de l'aplicació.....	57
4.6. Disseny de la interfície	59

CAPÍTOL 5

5.1. Introducció.....	62
5.2. Descripció funcional	62
5.2.1. Background Subtraction.....	62
5.2.2. Morfologia	64
5.2.3. Distància mínima entre rectangles	65

5.2.4. Agrupar.....	68
5.2.5. Kalman Filter	71
5.2.6. Compara Frames.....	73
5.2.7. Procés	78
 CAPÍTOL 6	 81
6.1. Introducció.....	82
6.2. Objectius	82
6.3. Planificació temporal final	84
6.4. Futures ampliacions i millores.....	85
6.5. Valoració personal	86
 Referències	 87
Agraïments	91

CAPÍTOL 1

Introducció

1.1. Motivacions

Les motivacions que han fet possible aquest projecte no són les típiques que solen tenir els projectes finals de carrera informàtics. Normalment, la idea ve donada pel director del projecte, el qual posa uns objectius i unes pautes que l'alumne va seguint fins a assolir-los utilitzant els coneixements assolits durant la carrera.

En aquest cas ha estat diferent, la motivació a seguir era aplicar allò après a l'assignatura Visió per Computador impartida a la carrera, però aprofundint en alguna de les seves vessants i, a partir d'aquí, decidir un projecte on l'objectiu era aprendre.

1.2. Objectiu

L'objectiu principal del projecte és, com s'ha dit al punt anterior, aprendre a utilitzar les tècniques d'una de les vessants de la Visió per Computador. En concret, el seguiment d'objectes.

Per tant, el projecte realitzat s'ha d'adaptar a l'objectiu principal. En un principi es va plantejar fer una aplicació, la funció de la qual fos un sistema de videovigilància intel·ligent, és a dir, a partir d'una càmera connectada a un ordinador i un recinte tancat, saber la gent que hi ha en tot moment, controlar on es dirigeixen, si entren o surten o detectar un comportament sospitos.

No obstant, aquest projecte era molt ambiciós i ja es va descartar la seva viabilitat a l'inici. Per això, es va decidir fer un projecte diferent que s'adaptés al temps i als recursos disponibles. Aquest és semblant a l'anterior però es queda a una capa inferior, més teòrica, i no aprofundeix tant en un tipus de programa amb una interfície molt elaborada dirigida a un client final.

Així, l'objectiu d'aquest projecte és, a partir de varies seqüències d'imatges amb escenes diferents, donar solució al problema de seguiment dels objectes que apareguin en ella. S'ha realitzat mitjançant la creació d'un prototip d'aplicació.

1.3. Organització de la memòria

La memòria està dividida en sis capítols i un annex.

El primer capítol és la introducció al projecte, on s'expliquen les motivacions que l'han dut a terme i es fa una primera visió de l'objectiu plantejat.

El segon capítol conté l'estudi de viabilitat, el qual està dividit en vuit parts. Aquestes són: una introducció al estudi, s'explica com està la situació actual, es numeren els requisits del projecte, es plantegen alternatives i s'arriba a una solució, es fa una planificació temporal, s'avaluen els riscos, es fa un pressupost i finalment s'arriba a una conclusió de tot l'estudi que dona viabilitat al projecte.

El tercer capítol són els fonaments teòrics sota els quals està regit el projecte. S'han tractat quatre temes crucials: un breu resum sobre la Visió per Computador, que són les imatges binàries, com funciona el filtre de Kalman i, per últim, què és la sostracció del fons de la imatge.

El quart capítol tracta tota la part del disseny de l'aplicació. Conté una descripció del problema del seguiment d'objectes, l'elecció del paradigma de desenvolupament, una especificació de requeriments exhaustiva i el disseny modular i de la interfície.

El cinquè capítol s'explica el desenvolupament del sistema i bàsicament té la descripció funcional dels mòduls més importants. A més a més, es comenten les traves trobades a l'hora de dur a terme la implementació.

El sisè i últim capítol conté les conclusions finals del projecte. On es comparen els objectius inicials amb els assolits, també es verifica si s'ha seguit la planificació temporal inicial, es parla de possibles ampliacions i millores futures del projecte i, finalment, es fa una valoració personal.

Al final de la memòria, hi ha un annex amb el manual d'usuari de l'aplicació.

CAPÍTOL 2

Estudi de viabilitat

2.1. Introducció

Es pretén fer una anàlisi que permeti avaluar les garanties d'èxit de la realització d'aquest projecte.

Inclou un estudi que proposa diverses solucions conceptuals al problema i, finalment, es tria una solució. A més a més, es defineixen els requisits del projecte, es fa una planificació d'aquest, s'avaluen riscos, es fa un pressupost, una anàlisi de costos i beneficis i, per acabar, una conclusió sobre si el projecte és viable o no.

2.1.1. Tipologia i paraules clau

- **Tipologia:** Projecte informàtic de desenvolupament d'un prototipus d'un sistema de software. No aplicable a distribució i basat en recerca.
- **Paraules clau:** Visió per Computador, seguiment d'objectes, detecció de moviment, intel·ligència artificial, webcam, filtre de Kalman, MATLAB.

2.1.2. Descripció

Dissenyar un prototip d'un sistema de detecció de moviment i seguiment d'objectes que tracti seqüències d'imatges provinents tant d'una webcam en temps real com de vídeos guardats.

2.1.3. Objectius del projecte

Com que aquest projecte es basa en crear un prototipus i no pas en crear un producte que, finalment, serà donat a un client, es plantegen els objectius de forma diferent.

El propi problema marca què s'ha de fer, es considera que l'objectiu principal és **l'aprenentatge**. La resta d'objectius són:

O.1. Entendre i aprendre a utilitzar algunes de les tècniques de Visió per Computador relacionades amb el tractament de vídeo.

O.2. A partir de diverses seqüències d'imatges, amb diferents escenes, detectar on hi ha moviment i fer un seguiment dels agents d'aquesta escena.

O.2.1. Seguiment de la trajectòria dels objectes en moviment.

O.3. Detectar quan apareix o desapareix un objecte a l'escena.

O.4. Distingir els objectes que hi ha dins de l'escena.

O.5. Obtenir informació estadística del procés.

Priorització dels objectius del projecte:

	<i>Crític</i>	<i>Prioritari</i>	<i>Secundari</i>
O.1.	X		
O.2.	X		
O.3.	X		
O.4.	X		
O.5.	X		

2.1.4. Definicions, acrònims i abreviacions

- Visió per Computador: és un subcamp de la intel·ligència artificial. Es tracta de programar un ordinador per a què aquest entengui una escena o les

característiques d'una imatge.

- Prototip: primera aproximació d'allò que després serà el software. No es cuiden detalls de la interfície gràfica sinó que els esforços es disposen en resoldre el problema en qüestió.
- Fals positiu: donar un resultat com a correcte sent incorrecte.
- Fals negatiu: donar un resultat com a incorrecte sent correcte.
- Temps real: és aquell sistema que interactua amb el món real i que dona una resposta correcta en un marge de temps concret.

2.1.5. Parts interessades

Stakeholders: Aquest projecte no en té ja que no es fabrica un producte per cap client, és a dir, no hi ha unitats organitzatives implicades en el projecte. Tot i així, si s'ha d'escollir un *stakeholder* seria el tutor del projecte Ramón Baldrich.

Perfils d'usuari: estarà enfocat a un usuari no expert que sigui capaç de fer servir l'aplicació, un cop llegit el manual, i que pugui entendre els resultats intuïtivament.

Project team:

Nom	Descripció	Responsabilitat
Ramón Baldrich	Cap del projecte (CP)	Defineix, gestiona, planifica i controla el projecte. S'encarrega de fer les aprovacions de les tasques realitzades.

David Perálvarez	Analista (A)	Fa l'estudi de viabilitat i la planificació. Analitza l'aplicació: arquitectura, metodologia, especificacions, estàndards.
David Perálvarez	Programador (P)	Dissenya i desenvolupa l'aplicació d'acord amb l'anàlisi i planificació prevista.
David Perálvarez	Tècnic de proves (TP)	Dissenya les proves i les realitza.
Ramón Baldrich	Director del projecte (DP) i/o tutor	Supervisa la feina de l'alumne i actua com <i>stakeholder</i> .

2.1.6. Producte i documentació del projecte

- a) Es lliurarà un prototip¹ d'aplicació informàtica.
- b) S'elaborarà una memòria del projecte.
- c) Es lliurarà un manual d'usuari inclòs com annex al final de la memòria.

2.2. Estudi de la situació actual

2.2.1. Context

En el cas d'aquest projecte no existeix un sistema actual ja creat que s'hagi de millorar o modificar. Es vol tractar el problema plantejat partint des de zero. Per tant, es crearà tot un sistema nou.

Primer es començarà tractant vídeos amb escenes molt senzilles, com pot ser una pilota en moviment en un fons controlat. Un cop dominades les tècniques que permeten el seguiment d'un objecte en moviment, es passarà a escenes

¹ En forma de codi, no d'executable. Caldrà tenir la versió de MATLAB utilitzada en el desenvolupament per poder-ho executar i utilitzar l'aplicació.

més complexes (on el fons no sigui controlat) i amb més d'un objecte en moviment. I, per acabar, es tractarà el problema de seguir diversos objectes en moviment en temps real, és a dir, que les imatges es capten al moment amb la webcam i es processen a l'instant.

2.2.2. Normatives i legislació

- a. Llei Orgànica 1/1982, de Protecció Civil del Dret a l'Honor, a la Intimitat Personal i Familiar i a la Pròpia Imatge.
- b. Normativa de projectes de final de carrera de la UAB.
- c. Llei de propietat intel·lectual.

2.3. Requisits del projecte

2.3.1. Requisits funcionals

- RF.1.** Funcionar amb vídeo i temps real.
- RF.2.** Detectar Background i Foreground.
- RF.3.** Detectar moviment dels objectes.
- RF.4.** Detectar posició dels objectes.
- RF.5.** Detectar aparició i desaparició dels objectes.
- RF.6.** Diferenciar objectes de l'escena.
- RF.7.** Seguiment de les trajectòries dels objectes.
- RF.8.** Mostrar resultats.

2.3.2. Requisits no funcionals

- RNF.1.** Compliment de la Llei Orgànica 1/1982 pel que fa referència al tractament de seqüències d'imatges on apareguin persones.

RNF.2. La visió dels resultats ha de ser intuïtiva.

RNF.3. Bon rendiment: ha de tenir un temps de resposta raonable.

RNF.4. Tolerància a errades i a accions incorrectes.

RNF.5. Evitar en la mesura del possible falsos positius i falsos negatius.

2.3.3. Restriccions del sistema

- a) L'aplicació ha de funcionar en un entorn Windows.
- b) L'aplicació ha d'adaptar-se a les necessitats.
- c) El projecte ha d'estar finalitzat abans del 30 de juny de 2010.

2.3.4. Catalogació i priorització dels requisits

Prioritats dels requisits funcionals:

	<i>RF. 1</i>	<i>RF. 2</i>	<i>RF. 3</i>	<i>RF. 4</i>	<i>RF. 5</i>	<i>RF. 6</i>	<i>RF. 7</i>	<i>RF. 8</i>
<i>ESSENCIAL</i>	X	X	X	X	X	X	X	X
<i>CONDICIONAL</i>								
<i>OPCIONAL</i>								

Prioritats dels requisits no funcionals:

	<i>RNF1</i>	<i>RNF2</i>	<i>RNF3</i>	<i>RNF4</i>	<i>RNF5</i>
<i>ESSENCIAL</i>	X	X	X		
<i>CONDICIONAL</i>				X	X
<i>OPCIONAL</i>					

Relació entre els requisits i objectius:

	<i>RF1</i>	<i>RF2</i>	<i>RF3</i>	<i>RF4</i>	<i>RF5</i>	<i>RF6</i>	<i>RF7</i>	<i>RF8</i>	<i>RNF1</i>	<i>RNF2</i>	<i>RNF3</i>	<i>RNF4</i>	<i>RNF5</i>
O1		X	X	X	X	X	X						X
O2	X	X	X	X			X	X	X	X	X	X	X
O3	X	X	X		X			X		X	X	X	X
O4	X	X		X		X		X		X	X	X	X
O5								X		X			

2.4. Alternatives i selecció de la solució

2.4.1. Alternativa 1

Realitzar l'aplicació mitjançant la creació, a partir de zero, de mètodes i algorismes de Visió per Computador per donar solució al problema.

- **Característiques:**

- Total independència de l'aplicació.
- S'ajusta als requisits que s'han definit.
- Ajustable als recursos disponibles.

- **Costos:**

- Requereix molt temps per un estudi previ exhaustiu de la matèria.

2.4.2. Alternativa 2

Realitzar l'aplicació mitjançant la combinació de mètodes i algorismes ja inventats (i només crear-ne els necessaris) de Visió per Computador per donar solució al problema.

- **Característiques:**

- Total independència de l'aplicació.

- S'ajusta als requisits que s'han definit.
- Ajustable als recursos disponibles.

- **Costos:**

- Segons la planificació prevista.

2.4.3. Alternativa 3

Adaptar una aplicació ja existent. Es tractaria d'estudiar-la, comprendre com està codificada i intentar crear una de pròpia amb les necessitats plantejades pel problema.

- **Característiques:**

- Independència condicionada.
- Ajustable als recursos disponibles.

- **Costos:**

- Requereix que el codi fet per l'altre programador sigui portable, ja que si no seria difícil adaptar-lo al problema plantejat.

2.4.4. Solució proposada

	<i>Costos adquisició</i>	<i>Costos adaptació</i>	<i>Nous recursos</i>	<i>Suport</i>	<i>Nivell integració</i>	<i>Complexitat</i>	<i>Formació</i>
<i>Alternativa 1</i>	0€	Baixos	No cal	Incloua en el projecte	Alt	Alta	Es desconeix
<i>Alternativa 2</i>	0€	Baixos	No cal	Incloua en el projecte	Alt	Mitjana	Incloua en el projecte
<i>Alternativa 3</i>	0€	Es desconeix/Alts	No cal	Web	Baix	Mitjana	Es desconeix

S'ha triat l'**alternativa 2** perquè s'adapta millor a la solució del problema, requereix menys temps que l'alternativa 1 i no té molt sentit que, avui en dia, amb el nombre tan elevat d'algorismes i tècniques de Visió per Computador

eficients que ja han estat inventades s'intentin crear de noves. Una altra raó per la qual s'ha escollit aquesta alternativa ha estat per complir un dels objectius d'aquest projecte: l'aprenentatge d'aquestes tècniques. L'alternativa 3 s'ha refusat. Primerament, perquè no s'ha trobat cap codi lliure relacionat amb el problema proposat i, en segon lloc, perquè es considera que costaria molt temps d'adaptació i no s'aprendria tant com amb l'alternativa 2.

2.5. Planificació del projecte

Calendari del projecte: El projecte es desenvoluparà de novembre de 2009 fins al maig de 2010 amb una dedicació de 12 hores per setmana. El total d'hores dedicades arriba a les 362 hores.

Data de començament: 1 de novembre de 2009

Data prevista de finalització: 31 de maig de 2010.

Eines de planificació i control: Microsoft Project.

2.5.1. Recursos del projecte

Recursos humans:

El projecte serà realitzat per dues persones: per una banda, Ramón Baldrich, el cap del projecte i, d'altra banda, David Perálvarez, l'analista, dissenyador, programador, tècnic de proves i formador.

És un projecte sense ànim de lucre i, per tant, ningú cobra per realitzar-lo. No obstant, es vol fer un pressupost fictici. Aleshores, podem dir que Ramón Baldrich cobraria 40€/h i David Perálvarez, 20€/h.

Recursos materials:

Els recursos necessaris per desenvolupar aquest projecte són bàsicament tres. Un ordinador, una webcam i el software MATLAB per implementar-ho. A més a més, s'utilitzarà per fer la documentació el Microsoft Office 2007 i el Microsoft Project 2003.

Característiques específiques dels recursos utilitzats:

- **Portàtil HP Pavilion dv6-2030ss:**
 - Windows 7
 - Processador AMD Athlon II Dual-Core
 - 320 GB de disc dur
 - 6 GB Memòria Ram
- **Ordinador de sobretaula APD:**
 - Windows XP Professional Service Pack 3
 - Intel® Core(TM)2 Quad CPU
 - 185 GB de disc dur
 - 3,49 GB Memòria Ram
- **Webcam Hercules Dualpix Chat and Show**
 - USB 2.0
 - Vídeo amb resolució màxima 1280x1024
 - Zoom 3x
 - Sensor CMOS
 - Fins a 30 fps (imatges capturades per segon)
- **MATLAB r2009a**

2.5.2. Tasques del projecte

Nº	Descripció de l'activitat	Durada	Pred.
1	Inici del projecte: assignació i matriculació del projecte	2h	
2	Planificació	36h	
3	Estudi de viabilitat	35h	1
4	Aprovació de l'estudi de viabilitat (Punt de control)	1h	3
5	Anàlisi de l'aplicació	60h	
6	Estudi previ de la matèria	36h	4

7	Arquitectura del sistema	4h	6
8	Anàlisi de requeriments	14h	7
9	Anàlisi de la seguretat	3h	8
10	Anàlisi legal	2h	9
11	Aprovació de l'anàlisi (Punt de control)	1h	10
12	Disseny de l'aplicació	19h	
13	Disseny d'interfícies	4h	11
14	Disseny modular de l'aplicació	4h	13
15	Disseny de tests	4h	14
16	Restriccions de software o hardware	2h	15
17	Restriccions de seguretat	2h	16
18	Compliment de normatives i legislació	2h	17
19	Aprovació del disseny (Punt de control)	1h	18
20	Execució	161h	
21	Selecció de metodologia de desenvolupament	2h	19
22	Selecció i adquisició de recursos	2h	21
23	Preparació de l'entorn de desenvolupament	2h	22
24	Construcció i desenvolupament de l'aplicació	155h	23
25	Control de qualitat	31h	
26	Tests unitaris	12h	24
27	Tests d'integració	12h	26
28	Tests d'estrès	6h	27
29	Aprovació d'execució i proves (Punt de control)	1h	28
30	Documentació	48h	
31	Elaboració de la memòria	44h	29
34	Manuais d'usuari	4h	32
35	Implantació i finalització del projecte	4h	33
36	Defensa del projecte	1h	34

2.5.3. Planificació temporal inicial

Diagrama de Gantt amb la planificació temporal del projecte:

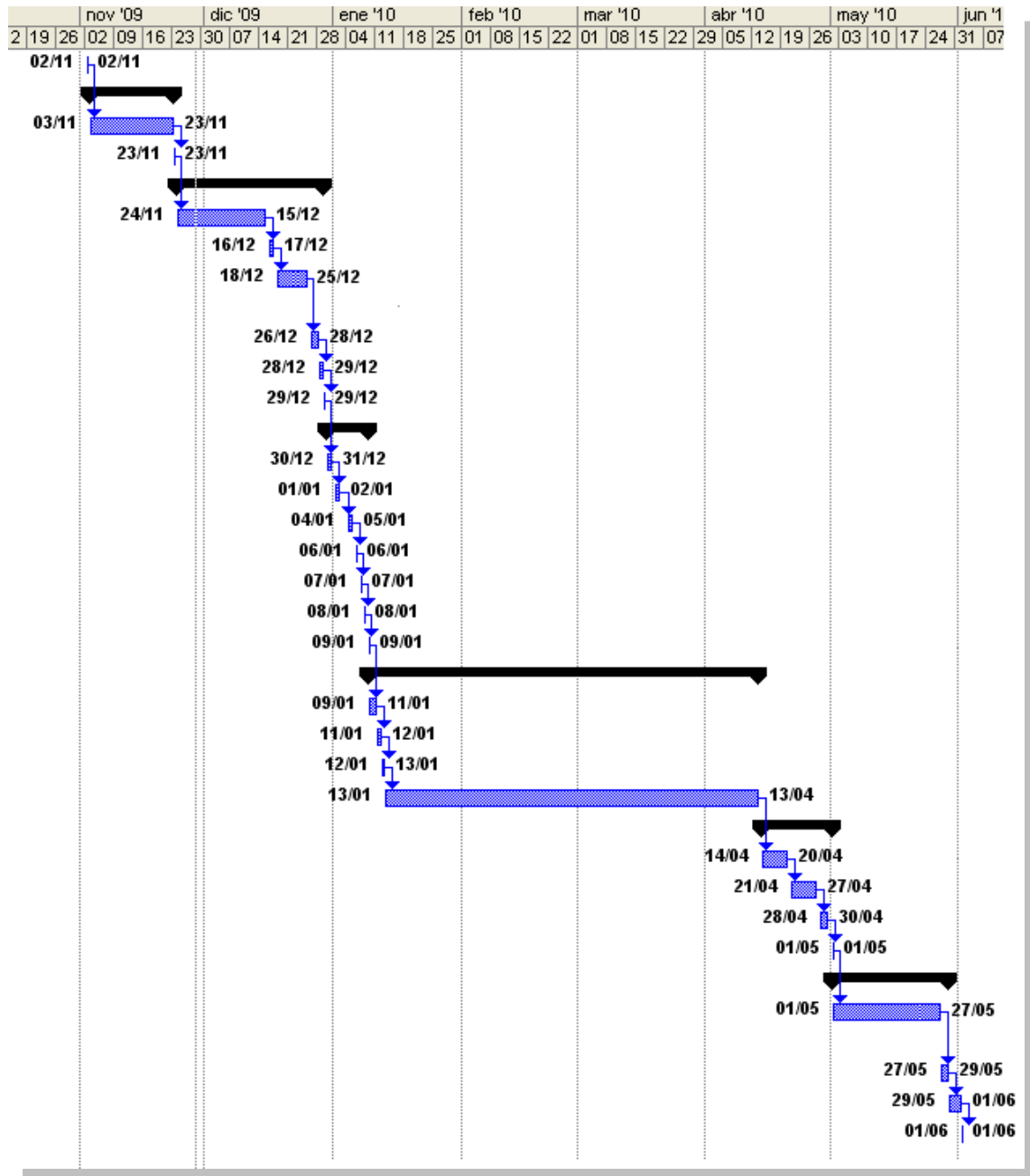


Figura 1: Diagrama de Gantt.

2.6. Avaluació de riscos

2.6.1. Llista de riscos

S'especifica el risc que pot ocórrer, la part concreta del projecte que quedaria afectada i la conseqüència d'aquest risc.

- R1.** Planificació optimista: estudi de viabilitat. No s'acaba a la data prevista.
- R2.** Nivell de dificultat dels objectius molt elevat: estudi de viabilitat. No s'assoleixen tots els objectius.
- R3.** Canvi de requisits: estudi de viabilitat, anàlisi. Endarreriment en el desenvolupament i resultat.
- R4.** No es realitza correctament la fase de test: desenvolupament, implantació. Manca de qualitat, deficiència en l'operativa.
- R5.** Incompliment de la llei o normativa del projecte: en qualsevol fase. Repercussions legals, incompliment d'objectius.
- R6.** Mala elecció de les tècniques de Visió per Computador: desenvolupament. No compleix els objectius.
- R7.** Algorismes no donen resposta en el temps adequat: desenvolupament, test. No compleix requeriment no funcional.
- R8.** Abandonament del projecte: en qualsevol fase. Frustració.

2.6.2. Catalogació de riscos

	<i>Probabilitat</i>	<i>Impacte</i>
<i>R1</i>	Alta	Crític
<i>R2</i>	Mitjana	Crític
<i>R3</i>	Baixa	Catastròfic

R4	Mitjana	Crític
R5	Baixa	Crític
R6	Mitjana	Crític
R7	Mitjana	Crític
R8	Baixa	Catastròfic

2.6.3. Pla de contingència

	<i>Solucions que cal adoptar segons els riscos</i>
R1	Ajornar alguna activitat, allargar data d'entrega.
R2	Negociar possibles canvis d'objectius a l'etapa d'estudi de viabilitat.
R3	Depenent de la fase del projecte on s'estigui, negociar que sigui una pròxima actualització del projecte, si es troba en fases inicials modificar-ne la planificació i el pressupost.
R4	Dissenyar les proves amb antelació i anar passant tests amb freqüència cada cert temps.
R5	Revisar les lleis que afecten al projecte, consultar si cal un expert, afrontar possibles repercussions penals.
R6	Investigar altres tècniques existents amb la mateixa funcionalitat i modificar la planificació.
R7	Revisar el codi i intentar optimitzar-lo evitant implementacions que gastin molts recursos.
R8	No té solució.

2.7. Pressupost

2.7.1. Estimació cost de personal

Tal i com dicta la normativa, el projecte és individual i tot el treball de la llista de tasques serà desenvolupat per David Perálvarez, excepte els punts de control, dels quals s'encarregarà Ramón Baldrich. Per tant, la taula de costos quedaria de la següent manera:

<i>Nom</i>	<i>Hores treballades</i>	<i>Cost</i>
David Perálvarez	358h	7160€
Ramón Baldrich	4h	160€

o

Total = 7320€

2.7.2. Estimació cost dels recursos

Només ha calgut adquirir la webcam, ja que la resta de recursos ja es tenien. Això ha implicat que solament s'ha considerat com a cost dels recursos el preu d'aquesta.

	<i>Cost amortització</i>	<i>Cost unitari</i>	<i>Període amortització</i>	<i>Període d'utilització</i>
Amortització webcam	9,72€	50€	36 mesos	7 mesos

Total = 9,72€

2.7.3. Resum i anàlisi cost benefici

Cost de desenvolupament del projecte..... 7320

Cost d'amortització del material..... 9,72

Cost de manteniment..... 0

Total =7329,72€

Aquest total correspon a un pressupost fictici. A més a més, no s'obtindran beneficis econòmics ja que l'aplicació realitzada no té com a propòsit ser venuda a cap client, sinó que s'elaborarà un prototip per estudiar i aprendre sobre aquest camp de la Visió per Computador.

2.8. Conclusions de l'estudi de viabilitat

Com s'ha dit al punt 2.5.1, aquest projecte no tindrà cap benefici econòmic, per tant, no té sentit avaluar si és viable o no en aquest sentit.

Es considera que es poden assolir els objectius proposats amb el temps i recursos disponibles més l'esforç i motivació.

El benefici principal haurà estat l'aprenentatge d'aquest camp de la informàtica que, a l'enginyeria tècnica informàtica de sistemes gairebé no es tracta però que dóna les bases per a poder especialitzar-se en qualsevol dels camps de la informàtica.

En conseqüència, com que l'objectiu principal és l'aprenentatge es considera que aquest projecte és **viable**.

CAPÍTOL 3

Fonaments teòrics

3.1. Què és la Visió per Computador?

3.1.1. Introducció

La Visió per Computador, també anomenada Visió Artificial, és una vessant de la Intel·ligència Artificial i es basa en un conjunt de tècniques que tenen com a finalitat extreure propietats del món a partir d'un conjunt d'imatges ^[16].

Es tracta de programar un ordinador per a obtenir certa informació a partir d'una imatge ^[17].

"Un sistema de Visió Artificial actua sobre una representació d'una realitat que li proporciona informació sobre la intensitat, colors, formes, etc. Aquestes representacions acostumen a ser en forma d'imatges estàtiques, escenes tridimensionals o imatges en moviment." ^[19].

3.1.2. Interpretació de la imatge

A partir d'ara, quan es parli d'una imatge bidimensional, s'estarà fent al·lusió a una funció f on a cada coordenada (x,y) de la imatge li correspondrà un valor d'alguna propietat com pot ser la intensitat o el color, per exemple.

Aquesta representació es dona quan es capta una imatge mitjançant una càmera i es passa a un digitalitzador, el qual duu a terme la digitalització².

Matemàticament tenim:

$$f(x,y) = I_{m,n} \tag{1}$$

on f,x,y pertanyen a \mathbb{R} i I,m,n pertanyen a \mathbb{Z}

² Mostrejar els valors d'intensitat en un conjunt discret de punts ^[20].

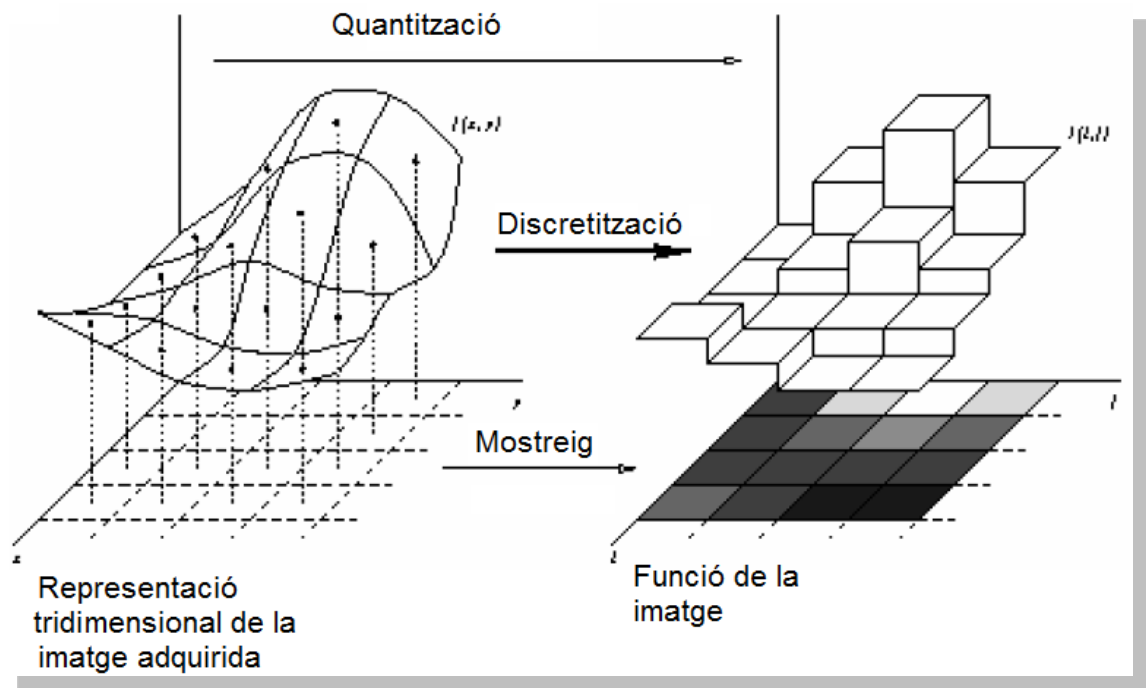


Figura 2: Procés de digitalització de la imatge.

Per tant, es pot definir una imatge discreta 2D com una matriu³ on cada píxel⁴ té un valor associat.

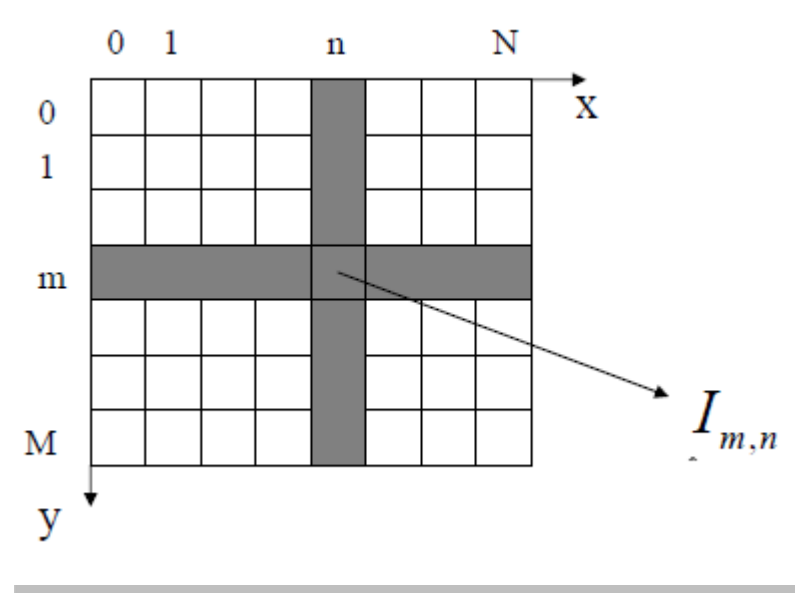


Figura 3: Matriu que representa una imatge 2D.

³ També anomenada reixeta o "grid".

⁴ És la unitat mínima d'una imatge 2D. Representa el valor d'intensitat en una posició determinada del grid.

3.1.3. Tipus d'imatges

3.1.3.1. Monocromàtiques

Hi ha dos tipus: les imatges en escala de grisos i les imatges binàries.

En el cas de les imatges en escala de grisos, els nivells d'intensitat van des de 0 (negre) a 255 (blanc).



Figura 4: Imatge en escala de grisos.

Les imatges binàries, en canvi, només tenen dos valors d'intensitat: o blanc (1) o negre (0).



Figura 5: Imatge binària.

3.1.3.2. Multiespectrals

La imatge està formada per la suma de varies bandes, on cada banda per si sola és una imatge monocromàtica. Hi ha diverses representacions de models de color, però la més coneguda és la RGB⁵.

⁵ Red, Green, Blue

En RGB associem a cada punt una terna de valors que indica la intensitat de cada canal (verd, vermell i blau).

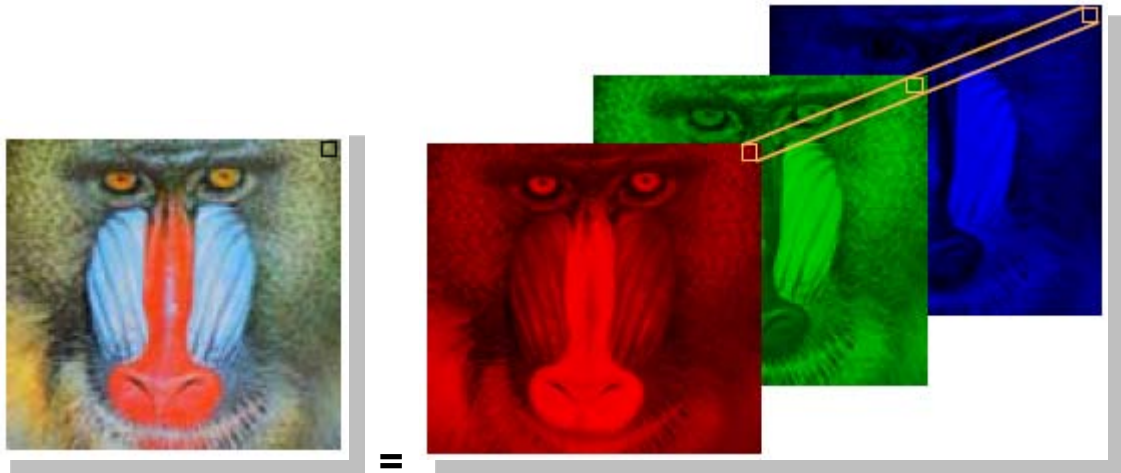


Figura 6: Com està formada una imatge en color.

3.1.4. Etapes d'un sistema de Visió Artificial

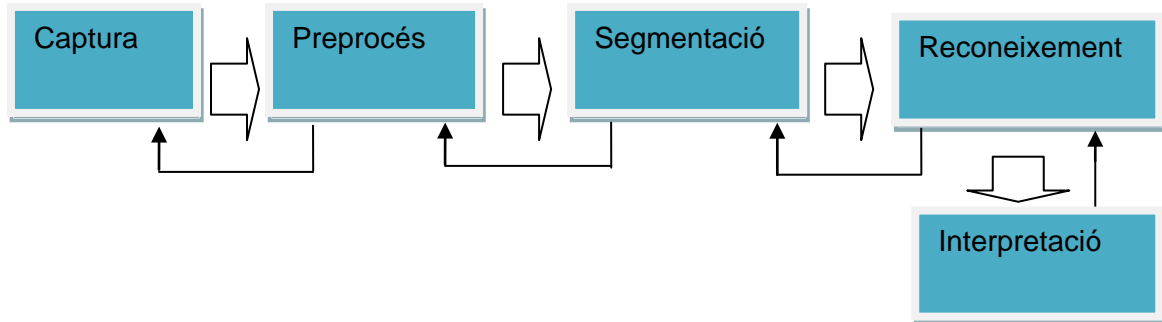
Un sistema de Visió per Computador està format per un sensor òptic, normalment una càmera més un computador, que emmagatzemarà les imatges i executarà els algorismes sobre elles.

El procés es pot dividir en cinc etapes:

1. Adquirir la imatge digital amb algun tipus de sensor.
2. Fer un preprocessament: es millora la qualitat de la imatge adquirida amb l'ajut de filtres i transformacions geomètriques per facilitar les etapes posteriors.
3. Segmentació: es tracta d'identificar els objectes de l'escena.
4. Reconeixement: poder diferenciar els objectes mitjançant l'ús de patrons⁶.
5. Interpretació: segons allò obtingut en el punt 4 es fa una interpretació i es prenen les decisions corresponents.

⁶ Anàlisi previ de certes característiques comunes que s'estableixen per diferenciar els objectes.

És un procés que pot realimentar-se cap endarrere, és a dir, pot ser que es faci una mala interpretació i, per tant, calgui tornar al reconeixement, segmentar la imatge d'una altra forma o passar-li un filtre diferent en el preprocés...^[19]



3.1.5. Aplicacions de la Visió Artificial

Alguns exemples que inclouen sistemes de Visió per Computador són:

1. Inspecció industrial: en comprovació de peces correctes, control de qualitat de medicaments o aliments.
2. Detecció d'esdeveniments: videovigilància, recompte de persones, seguiment d'objectes.
3. Interacció: fa d'entrada en dispositius d'interacció computador-humà. Un exemple podria ser un videojoc en què es juga mitjançant una webcam.
4. Control de processos: robots industrial o vehicles autònoms.

3.2. Imatges binàries

3.2.1. Introducció

Com s'ha mencionat al punt 3.1.3.1 les imatges binàries pertanyen al tipus d'imatges monocromàtiques.

En aquest punt es farà una definició formal de les imatges binàries, s'explicaran conceptes com el veïnatge, l'etiquetatge i altres que seran útils per entendre millor com s'han implementat els mòduls de l'aplicació.

3.2.2. Definició

Una imatge binària és una matriu de zeros i uns, on el zero vol dir que el píxel és negre i l'u, que és blanc:

$$I : N \times M \rightarrow \{0, 1\} \quad (2)$$

on I és la intensitat i N, M les dimensions de la matriu.

$$(x, y) \mapsto I(x, y) \quad (3)$$

on cada punt de la matriu li correspon un valor d'intensitat I .

3.2.3. Veïnatge

El veïnatge serveix per definir quants elements hi ha en una imatge. Si tenim un píxel (x, y) i està envoltat per la regió ^[23]:

8 (x-1,y-1)	1 (x,y-1)	2 (x+1,y-1)
7 (x-1,y)	0 (x,y)	3 (x+1,y)
6 (x-1,y+1)	5 (x,y+1)	4 (x+1,y+1)

Figura 7: Veïnatge d'un píxel.

Tindrem dos tipus de connectivitat:

1. Connectivitat a quatre: considerem veïns de 0 només els píxels horitzontals i verticals.
2. Connectivitat a vuit: considerem veïns de 0 els píxels horitzontals, verticals i diagonals, és a dir, tots els que l'envolten.

Per tant, es dirà que una regió està connectada si existeix un camí entre tots els píxels d'aquesta regió. Cada *regió connectada* serà considerada com un *objecte*.

3.2.4. Etiquetatge

L'etiquetatge es basa en assignar a cada píxel una etiqueta que indiqui l'objecte de la imatge al qual pertany ^[23].

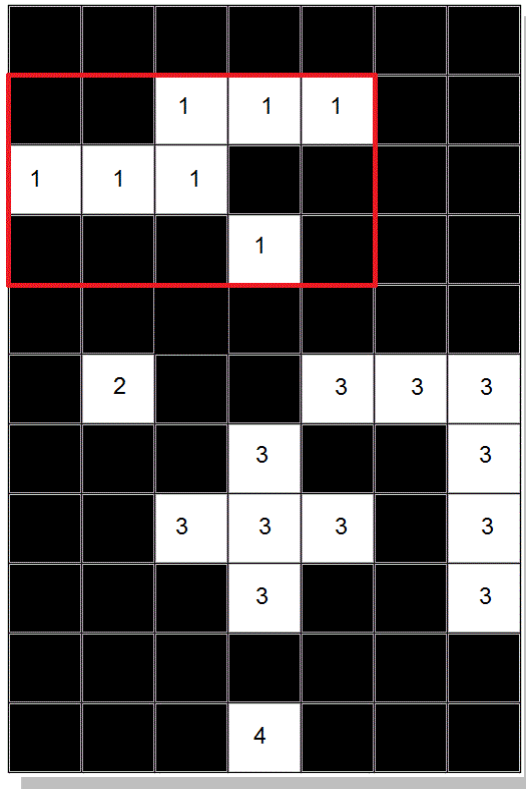


Figura 8: Representació d'una imatge binària etiquetada amb connectivitat a vuit.

Nota: Tots els píxels en negre estan etiquetats amb un zero.

Un cop tenim la imatge etiquetada és fàcil treure informació interessant com:

- *Nombre d'objectes*: es tracta de buscar el nombre màxim de l'etiquetatge, en aquest cas 4.
- *Àrea d'un objecte*: es calcula sumant el nombre de píxels que té l'objecte.
- *Centre de masses d'un objecte*: és el píxel de la imatge en què hi ha la mateixa quantitat d'objecte al seu voltant i es calcula:

$$C_x = \sum_{\text{píxels}} x_i / N \quad (4)$$

es sumen totes les coordenades x de l'objecte i es fa la mitjana.

$$C_y = \sum_{\text{píxels}} y_j / N \quad (5)$$

es sumen totes les coordenades y de l'objecte i es fa la mitjana.

Així el centre de masses d'un objecte qualsevol serà (C_x, C_y) .

- *Bounding Box*: és la caixa mínima que inclou un objecte, és a dir, el rectangle mínim on hi cap l'objecte. Té com a centre el centre de masses. En la figura anterior el rectangle vermell representa el Bounding Box de l'objecte 1.

3.2.5. Morfologia binària

A vegades, per poder obtenir una característica determinada d'un objecte qualsevol, cal que modifiquem la forma d'aquest. D'això s'encarrega la morfologia.

Concretament la morfologia binària considera que les imatges binàries són conjunts i, per tant, se'ls pot aplicar totes les operacions de la Teoria de Conjunts (com la unió, intersecció, etc.).

En aquest projecte només s'aplica una funció morfològica i es tracta de la **Erosió**, la qual es basa en "*Eliminar tots els píxels de la frontera dels objectes de la imatge*"^[23]. La utilitzem per esborrar brutícia en la part de disseny que correspon a Preprocessament.

La definició matemàtica de l'erosió és la següent:

$$\left\{ \bar{x} \mid B_{\bar{x}} \subset X \right\} \quad (6)$$

on X és la imatge binària i B és l'element estructurant⁷ amb el qual s'aplicarà la transformació. Cal observar l'exemple següent:

⁷ Objecte amb la forma que nosaltres volem que interactua amb una imatge amb el propòsit de canviar les formes d'aquesta imatge.

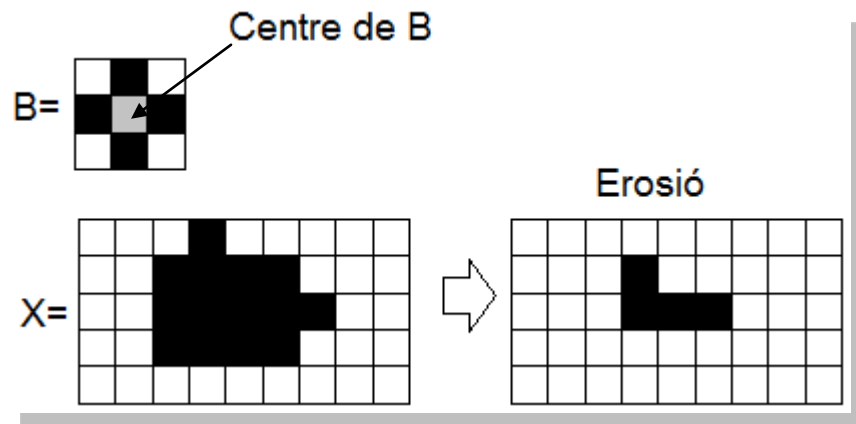


Figura 9: Exemple d'erosió.

3.3. Filtre de Kalman

3.3.1. Introducció

Degut a la importància que té aquest filtre en el projecte a continuació s'inclou un breu resum teòric sobre què és el filtre de Kalman.

L'any 1960 Rudolph E. Kalman va publicar una solució recursiva⁸ al problema del filtratge lineal de dades discretes. Des d'aquest moment, degut a l'avanç en la computació digital, el filtre de Kalman ha sigut motiu de recerca i aplicació, particularment en l'àrea de la navegació autònoma o assistida ^[14].

3.3.2. Descripció

Es parteix del model d'un procés, del qual es vol obtenir informació de la seva dinàmica, però no es poden mesurar les variables directament, sinó que es fa ús d'informació observada i coneixement previ per poder produir una estimació de les variables d'estat⁹. El problema ve donat quan s'utilitza aquest model amb mesures contaminades amb algun tipus d'incertesa. Això s'intenta resoldre amb el filtre de Kalman ^[10].

⁸ El terme recursiu aquí vol dir que el filtre de Kalman calcula de nou la solució cada cop que una nova observació o mesura s'incorpora al sistema.

⁹ Entenem per estat com la informació important d'un sistema en un cert instant de temps.

És una tècnica d'estimació Bayesiana que permet seguir sistemes estocàstics¹⁰ dinàmics i que es basa en un conjunt d'equacions matemàtiques capaces de proporcionar un estimador òptim d'aquest sistema, del qual disposem mesures al llarg del temps ^{[13][15]}.

Funciona mitjançant un mecanisme de predicció i correcció. És a dir, és capaç de predir el nou estat a partir de l'estimació prèvia, afegint un terme de correcció proporcional a l'error de predicció, el qual és minimitzat estadísticament ^[9].

3.3.3. Formulació del filtre de Kalman Discret

Es vol una estimació de l'estat \mathbf{x} , que pertany a \mathbb{R}^n , d'un procés controlat en temps discret descrit per a la següent *equació del procés*:

$$\mathbf{x}_{t+1} = \mathbf{A}_t \mathbf{x}_t + \mathbf{B} \mathbf{u}_t + \mathbf{w}_t \quad (7)$$

amb una mesura \mathbf{z} que pertany a \mathbb{R}^m que es tracta de l'observació i ve descrita per la *equació de la mesura*:

$$\mathbf{z}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t \quad (8)$$

El paràmetre \mathbf{A} de (7) és una matriu de dimensió $n \times n$ que relaciona l'estat en temps t amb l'estat en temps $t+1$. \mathbf{H} de (8) és una matriu de dimensió $m \times n$ i relaciona l'estat amb la mesura \mathbf{z} . Ambdós es consideren constants al llarg del temps.

Els paràmetres \mathbf{w} de (7) i \mathbf{v} de (2) són variables aleatòries que representen l'error del procés i l'error de la mesura respectivament, i es consideren soroll blanc i independents entre elles. A més, tenen una distribució normal:

$$p(\mathbf{w}) \approx N(0, \mathbf{Q}) \quad (9)$$

$$p(\mathbf{v}) \approx N(0, \mathbf{R}) \quad (10)$$

¹⁰ En l'àmbit de la intel·ligència artificial una cosa estocàstica vol dir que funciona mitjançant mètodes probabilístics.

on Q és la matriu de covariància de la pertorbació del procés¹¹ i R és la matriu de covariància de la pertorbació de la mesura¹² les quals s'assumeix que són constants per simplicitat.

3.3.4. Ús del filtre en la Visió per Computador

En Visió per Computador s'utilitza en el seguiment d'objectes en seqüències d'imatges ja que és tracta d'un sistema dinàmic¹³ representat en la forma estat-espai¹⁴. Cal afegir que els sensors de les càmeres estan sotmesos al soroll de captació. Per tant, el filtre de Kalman serà una bona solució.

L'algorisme s'utilitza per estimar la posició d'un component¹⁵ en moviment en la següent imatge. És a dir, es basa en buscar el component en una àrea concreta de la següent imatge al voltant de la posició que s'ha predit, en la qual s'està segur que es trobarà el component amb un grau de confiança^[13].

Per poder aplicar les funcions anteriors en el cas del seguiment d'objectes, prèviament s'han de considerar diversos aspectes. L'estat " X " es correspon amb el vector de posició de l'objecte " x_x ", " x_y " en la imatge i amb el vector de velocitat " v_x ", " v_y ". Per últim, l'observació " z " és un vector de posició observada de l'objecte " z_x ", " z_y ".

Un cop es sap això, es pot veure fàcilment d'on surt la matriu " A ". El següent sistema d'equacions ho mostra:

$$\left. \begin{aligned} X_{xt+1} &= X_{xt} + V_{xt} \\ X_{yt+1} &= X_{yt} + V_{yt} \\ V_{xt+1} &= V_{xt} \\ V_{yt+1} &= V_{yt} \end{aligned} \right\} \quad (11)$$

¹¹ També anomenada matriu de covariància del soroll intrínsec.

¹² També anomenada matriu de covariància del error mesurat.

¹³ Considerem que el moviment en l'escena és continu, per tant, es podran fer prediccions sobre el moviment d'objectes durant el temps.

¹⁴ El sistema està descrit per un conjunt de variables d'estat.

¹⁵ Ja sigui un punt, regió, contorns, cantonada, etc. de l'objecte.

La primera i segona equació del sistema volen dir que la posició x, y de l'estat X en temps $t+1$ és igual a la posició en temps t sumant-li una velocitat v_x, v_y respectivament. La quarta i última equació indiquen que s'assumeix que les velocitats són constants (no hi ha acceleració). Ja només falta expressar aquest sistema en forma de matriu i s'obté "A"¹⁶:

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matriu B relaciona l'entrada control u amb l'estat x i la matriu H l'estat amb la mesura z :

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

3.3.5. Algorisme

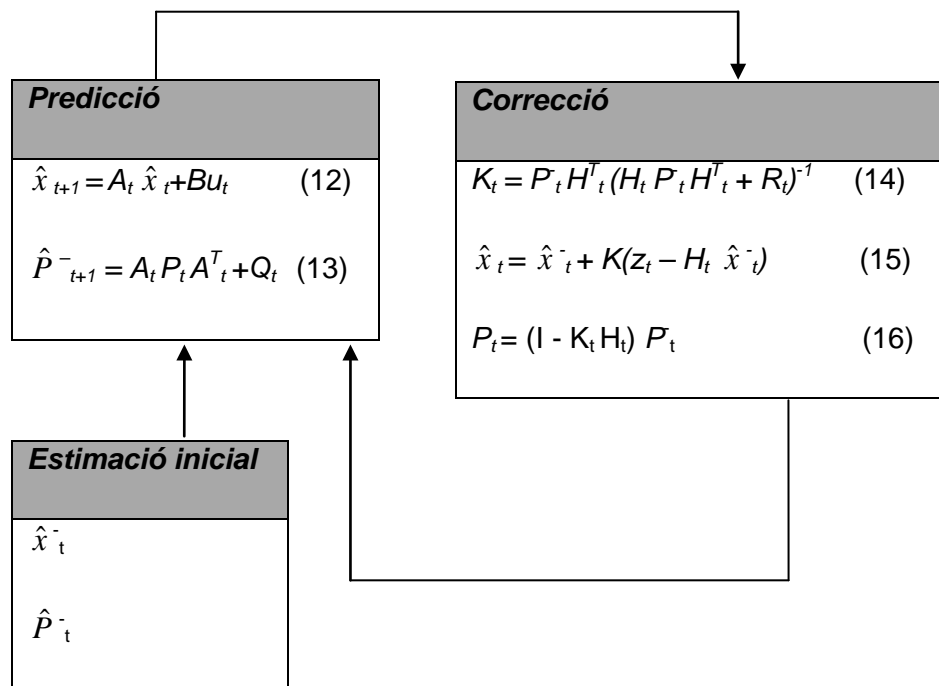
Les equacions de l'algorisme del filtre de Kalman es poden dividir en dos grups: *predicció* i *correcció*.

El primer grup, són les que actualitzen el temps, és a dir, projecten l'estat a temps $t+1$, prenent com a referència l'estat en temps $t-1$, i fan una actualització intermèdia de la matriu de covariància de l'estat, obtenint així els estimadors a priori.

En el segon, estan les que actualitzen les dades observades, són responsables de la retroalimentació, amb la qual obtenim nova informació de l'estimació anterior aconseguint així una estimació millorada a posteriori. Per tant, l'algorisme primer fa una predicció i després la corregeix ^[9].

Amb el següent esquema es vol mostrar les equacions de cada grup i el funcionament del filtre:

¹⁶ Com es veurà a l'apartat "Descripció del problema del seguiment d'objectes" al capítol de "Disseny de l'aplicació" la matriu A descriu el model dinàmic del sistema.



Definicions:

\hat{x}_t^- : Estimador de l'estat x a priori.

\hat{x}_t : Estimador de l'estat x a posteriori.

z_t : Observació actual.

$H_t \hat{x}_t^-$: Predicció de la mesura.

$(z_t - H_t \hat{x}_t^-)$: Innovació de la mesura.

\hat{P}_t^- : Estimador de la covariància de l'error a priori.

K : Guany de Kalman.

Funcionament:

En el requadre de predicció, l'equació (12) projecta cap endavant l'estimador de l'estat, i la (13) fa un pronòstic de la covariància de l'error a priori.

En el requadre correcció, l'equació (14) calcula el guany de Kalman, el qual, és una matriu que estableix com afectarà l'error entre la mesura i l'estimació. L'equació (15) actualitza l'estimació de l'estat amb la mesura. Hi ha dos casos límits: en primer lloc, si R_t tendeix a 0 en l'equació (14) vol dir tenir més confiança en la innovació de la mesura en l'equació (15) i, en segon lloc, si P_t tendeix a 0 en l'equació (14) vol dir que tindrem més confiança en la mesura que hem predit. Per últim, l'equació (16) actualitza la covariància de l'error a posteriori.

Per acabar, l'explicació del filtre de Kalman ressalta que l'estimació inicial de \hat{x}_t i \hat{P}_t no és massa crítica¹⁷. A més a més, el valor de R es pot estimar mitjançant les mesures, en canvi, Q no es pot estimar perquè no hi ha accés directe a l'estat. En ambdós, normalment s'obté el seu valor realitzant un ajust¹⁸ [12].

3.4. Sostracció del fons de la imatge

La sostracció del fons, també anomenada *Background Subtraction*, es basa en eliminar d'una seqüència d'imatges tot allò que no estigui en moviment, per així obtenir imatges en les quals només hi hagi objectes en moviment, anomenades *Foreground*.

Avui en dia existeixen diferents mètodes, tots ells amb avantatges i flaqueses, ja que poden ser molt eficients a nivell computacional, però no obtenen un resultat òptim, i a l'inrevés, fent-los així ineficients per aplicacions en temps real. L'objectiu, per tant, és aconseguir que aquests mètodes siguin eficients computacionalment i s'obtingui un bon resultat per un posterior tractament en l'aplicació. En aquest projecte serà vital poder extreure el fons de les imatges per obtenir els objectes en moviment i poder fer el seu seguiment.

¹⁷ No ens influirà en un bon o mal comportament del filtre.

¹⁸ Es proven diversos valors i es tria el que doni un bon resultat.

S'han estudiat tres mètodes diferents, els quals apareixen enumerats de forma ascendent segons la seva dificultat de comprensió ^[22]:

1. Mètode de diferència de frame (*frame difference*).
2. Mètode de la mediana (*median method*).
3. Mètode de combinació de Gaussianes (*Gaussian mixture*).

Per fer una selecció del mètode utilitzat a l'aplicació s'han tingut en compte les avantatges i inconvenients de cadascun ¹⁹:

Mètode	Avantatges	Inconvenients
Frame difference	<ul style="list-style-type: none"> - Molt baix pes computacional. - El model de Background creat és altament adaptatiu. - Extreu bé soroll estrany de l'escena. 	<ul style="list-style-type: none"> - Per objectes amb intensitats uniformement distribuïdes fa que detecti els interiors dels objectes com a Background.
Median method	<ul style="list-style-type: none"> - Separa bé l'objecte sense considerar Background el seu interior. - Té bons resultats que s'acosten als que es poden obtenir amb un mètode d'alta complexitat. 	<ul style="list-style-type: none"> - Pot deixar restes d'ombres o rastres dels objectes en moviment. - Requereix un cost computacional més elevat. - Requereix més memòria per construir el model del Background.
Gaussian mixture	<ul style="list-style-type: none"> - Bo separant objectes. - Bo eliminant soroll estrany de l'escena. - Bo eliminant ombres. 	<ul style="list-style-type: none"> - És molt sensible als canvis d'il·luminació. - Necessita una inicialització òptima dels seus paràmetres per a un bon funcionament.

Tots tres tenen l'inconvenient de què necessiten un bon valor de *Threshold* ²⁰ per oferir bons resultats i aquest valor no ve preestablert, sinó que s'ha de decidir.

¹⁹ Encara que aparegui una taula amb els avantatges i desavantatges de cadascun dels tres mètodes, posteriorment només serà explicat en profunditat el mètode escollit.

Després de comparar els tres mètodes, l'elegit ha estat el segon. Concretament, una aproximació al segon. Per això se li dirà *Aproximate Median Method*.

El mètode original necessita molta memòria, ja que per calcular el seu model de Background ha d'emmagatzemar els N previs frames i calcular la mediana de tots ells. En canvi, el mètode aproximat no necessita emmagatzemar cap frame previ per construir el seu model perquè ho fa de la següent forma:

Es parteix d'un model de Background²¹ inicial. Amb aquest model es fa una comparació píxel per píxel amb cada frame de la seqüència d'imatges. Si un píxel del frame actual té un valor d'intensitat més gran que el mateix píxel del model Background, s'incrementa amb 1 el valor del píxel del Background. En canvi, si un píxel del frame actual té un valor menor se li resta 1. D'aquesta manera es va actualitzant el model.

El funcionament de la sostracció és el mateix que el del mètode *Frame difference*. És a dir, a cada imatge se li sostrau el model de Background i cada valor obtingut d'intensitat es compara amb el valor del Threshold per decidir si el píxel pertany al Foreground o no:

$$|bg - f_n| > T \quad (17)$$

on bg és el model de Background, f_n és el frame actual i T és el valor del Threshold.



Figura 10: Frame actual, el model de Background i Foreground respectivament .

²⁰ És el llindar de detecció de canvi, és a dir, és un valor límit d'intensitat per distingir si un píxel de la imatge pertany al Background o al Foreground.

²¹ És una imatge que representa tot allò de la imatge que no tingui moviment, que és comparada amb la seqüència d'imatges d'entrada per així poder sostreure'ls els fons.

CAPÍTOL 4

Disseny de l'aplicació

4.1. Introducció

En aquest capítol, primerament es realitzarà una descripció del problema del seguiment per veure com influeix en el disseny. A continuació, es triarà un paradigma de desenvolupament. Seguidament, es durà a terme una especificació detallada dels requeriments del projecte, a més a més de mencionar les restriccions que haurà de tenir el sistema. Per últim, es detallarà el disseny modular que tindrà l'aplicació i el disseny de la interfície.

4.2. Descripció del problema del seguiment d'objectes

En el seguiment d'objectes, s'han de tenir en compte diverses consideracions que influiran a l'hora de dissenyar la nostra aplicació. Concretament són tres punts diferents: el model dinàmic, el comportament i l'evolució dels objectes.

4.2.1. Model dinàmic

Hi ha tres formes diferents de moviment d'un objecte:

1. **Moviment a velocitat constant:** es considera que els objectes en moviment no canvien la seva velocitat al llarg del temps. És a dir, no hi ha acceleració. Això vol dir que el mòdul, la direcció i el sentit de la velocitat de l'objecte no canvia al llarg del temps.
2. **Moviment amb acceleració constant:** és aquell moviment on la velocitat varia al llarg del temps, però de forma constant. És a dir, l'acceleració que s'aplica sobre l'objecte és constant en magnitud i direcció en tot el recorregut. Per exemple, un cotxe que varia la seva velocitat de la següent forma: a temps $t=1$ va a 10 Km/h, a $t=2$ va a 20 Km/h i a $t=3$ va a 30 km/h i així successivament. Per tant, l'acceleració serà constant i valdrà 10 km/h^2 .
3. **Moviment amb acceleració variable:** és la més complexa de les tres formes. En aquest tipus de moviment l'acceleració pot variar en

el temps. Per exemple, el mateix cotxe que abans, però ara a $t=1$ va a 10 Km/h, a $t=2$ va a 15 Km/h i a $t=3$ a 100 Km/h. Com es pot comprovar, l'acceleració no ha sigut constant.

El model dinàmic que s'ha decidit per l'aplicació ha estat el primer. Es considerarà a l'hora de fer el seguiment que els objectes tenen velocitat constant i no tenen acceleració. Això es veu reflectit en el paràmetre "A" del filtre de Kalman, que és la matriu que defineix el model dinàmic.

4.2.2. Comportament dels objectes

Es considera que la trajectòria que dibuixa un objecte quan es mou és una funció. Per tant, hi ha dos tipus de funcions:

1. Funcions contínuament diferenciables: això vol dir que la primera derivada està definida en tots els punts de la trajectòria que forma i, per tant, se li diu que és de la classe C^1 perquè les seves derivades parcials són contínues.

Un exemple seria la trajectòria que fa un cotxe sobre una carretera de corbes (és una trajectòria suau).

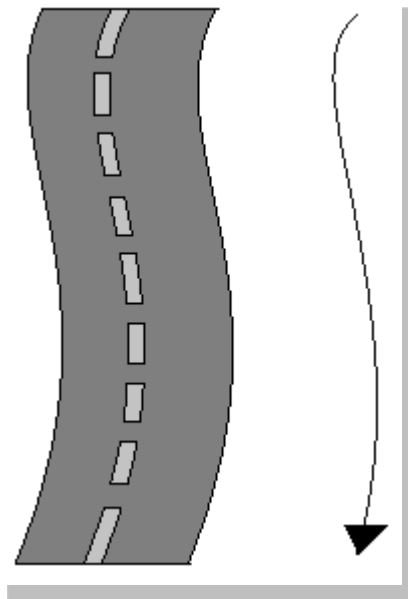


Figura 11: Trajectòria que faria un vehicle.

2. Funcions no contínuament diferenciables: es tracta de funcions on la primera derivada no està definida en tots els punts de la funció i pot presentar singularitats com ser infinita.

Pertanyen a aquest tipus els objectes que tenen trajectòries amb canvis de posició bruscos o estan sotmesos a rebots. Alguns exemples podrien ser: les boles d'un billar, una pilota de bàsquet que va rebotant o la trajectòria que fa una pilota en un partit de futbol:

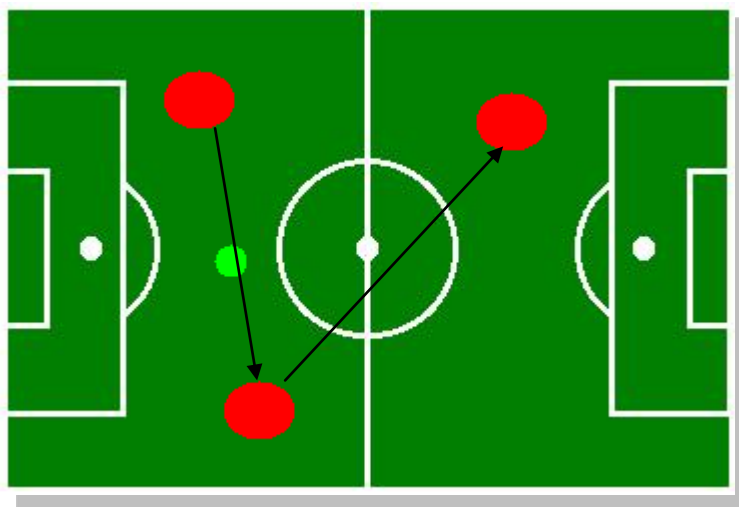


Figura 12: Trajectòria que faria una pilota en un partit de futbol.

A l'aplicació es buscarà solució pel comportament del primer cas. És a dir, objectes que segueixen trajectòries suaus. L'elecció d'aquest comportament ve marcat pel funcionament del filtre de Kalman, que en el segon cas no donaria una bona estimació, ja que són moviments aleatoris difícils de preveure i el que faria és retornar un valor que té més confiança en la innovació ja corregida que no pas en l'estimació²².

4.2.3. Evolució dels objectes

L'evolució dels objectes es pot entendre com l'estat en què es troben al llarg d'una seqüència:

- **Permanència de l'objecte:** vol dir que l'objecte està a l'escena i es pot

²² Per entendre-ho millor llegir l'apartat Fonaments teòrics on es parla del Filtre de Kalman.

veure com es mou.

- **Desaparició de l'objecte:** no es veu on està l'objecte. Hi ha dos casos:
 1. **Oclusió:** això passa quan l'objecte s'oculta un cert temps (normalment perquè es creua amb un objecte gran immòbil) i després reapareix de nou en l'escena.
 2. **Total:** es tracta d'una desaparició permanent. L'objecte no torna a reaparèixer més.
- **Fragmentació de l'objecte:** quan es duu a terme la segmentació de la imatge per obtenir els objectes que es volen seguir, pot ser que d'un frame al següent algun dels objectes es desfacin en diversos trossos a causa del resultat de la sostracció del fons.
- **Aparició de l'objecte:** succeeix quan a l'escena apareix un nou objecte, el qual s'ha de seguir.
- **L'objecte s'atura:** vol dir que l'objecte està a l'escena però no es mou.

Es tindran en compte tots aquests estats quan s'implementi la solució.

4.3. Paradigma de desenvolupament

A l'hora de desenvolupar el projecte s'ha hagut d'escollir un model de desenvolupament. El més adient és el model evolutiu.

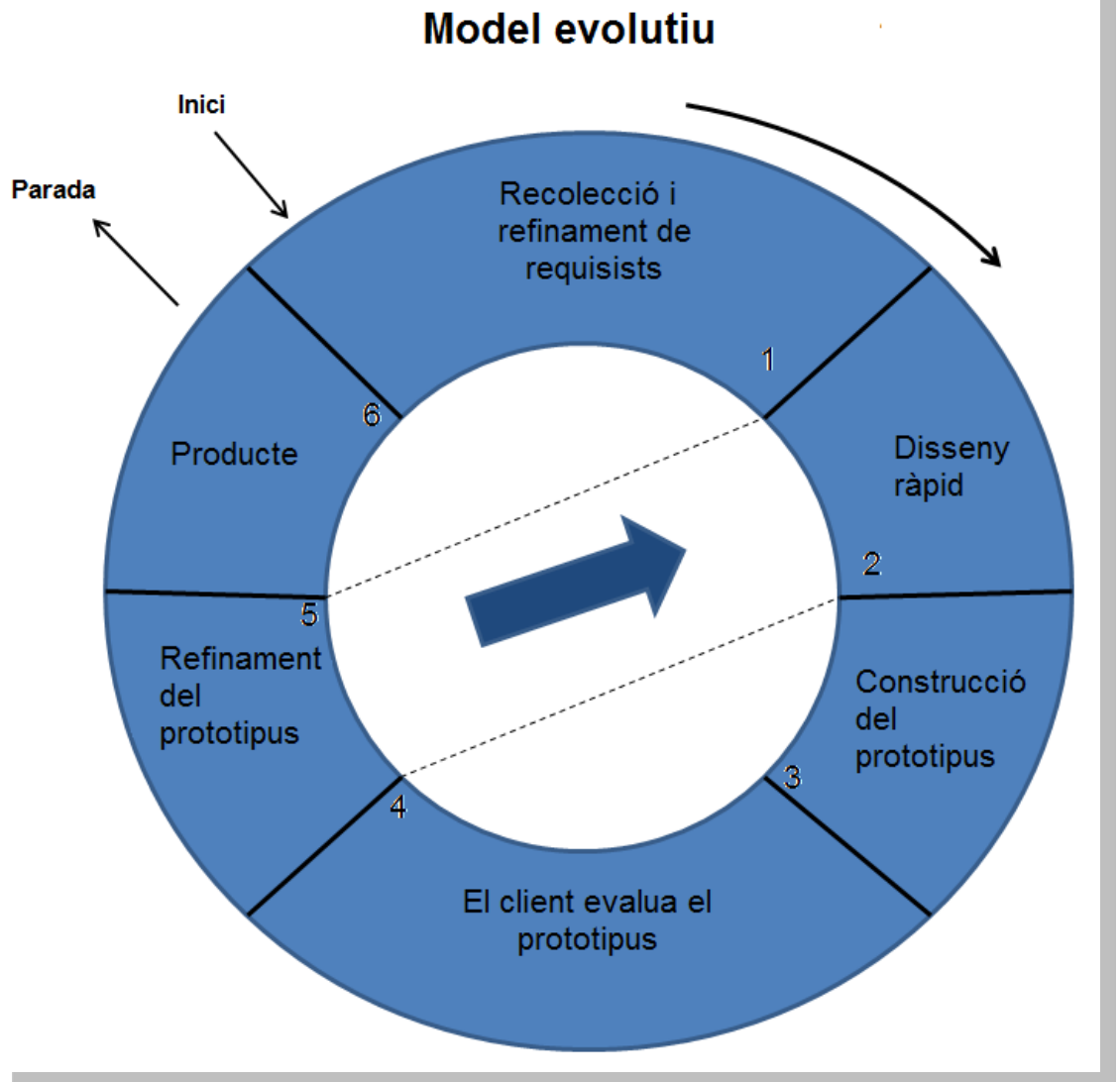


Figura 13: Funcionament del model evolutiu.

Aquest intenta combinar els beneficis que té el *model de prototipatge* amb el *model de vida clàssic*. El seu funcionament és el següent:

1. Es queda amb el client²³ per fer la recollida i anàlisi dels requeriments.

²³ En aquest cas, el client és el mateix projectista amb l'ajut del director del projecte. Per tant, ells són qui decideixen els requeriments.

2. Després es fa un disseny ràpid de l'aplicació segons els requeriments.
3. A partir del disseny es construeix de forma incremental i ràpida un prototip sense tenir molt en compte el disseny de la interfície. Per tant, cada cop s'afegeix una nova funcionalitat al sistema i es prova.
4. El prototip es presenta al client i aquest decideix quines coses cal afegir, així pot veure com va el seu projecte.
5. El projectista va refinant el prototip segons allò que hagi decidit amb el client.
6. En cas que apareguin nous requisits anar al punt 2, si no ja s'haurà acabat.

Avantatges:

- Mentre es va fent el projecte es poden anar fent reunions amb el director del projecte per mostrar-li resultats de com va evolucionant.
- El client pot prendre decisions immediates sobre els requisits.
- El manteniment es fa durant tot el procés i no al final.

Inconvenients:

- Evitar en la mesura del possible que apareguin nous requisits, ja que poden provocar que no s'acabi el projecte a la data fixada. Caldrà marcar un límit en la inclusió de requeriments.
- Pot tenir escalabilitat reduïda si el prototipus es vol transformar finalment en una aplicació real. En aquest cas no existeix aquest programa perquè l'objectiu és crear un prototipus. Si posteriorment es volgués crear una aplicació, caldria utilitzar altres llenguatges de programació més òptims per aplicacions de temps real i creació d'interfícies i, a més, aplicar un nou paradigma diferent.
- A l'hora de fer el prototipus no utilitzar recursos adequats degut a què es vol crear ràpidament.

4.4. Especificació de requeriments

*"**Requeriments:** conjunt d'idees que el client té sobre el que ha de ser el software a desenvolupar. Són les prestacions del sistema."* ^[7].

4.4.1. Requeriments funcionals

Els requeriments funcionals ens descriuen quin és el comportament que ha de tenir el software. Ens indiquen quines sortides s'han d'obtenir segons les entrades que hi hagi i les operacions que calguin.

En aquest apartat s'aprofundirà en l'explicació de cada un dels requeriments funcionals establerts a l'estudi de viabilitat.

- **Funcionar amb vídeo i temps real**

L'aplicació ha de ser capaç de fer tot el procés independentment si l'entrada és un vídeo o imatges captades, a temps real des d'una webcam. Als dos tipus d'entrada se'ls aplicarà el mateix tractament per aconseguir el seguiment dels objectes.

- **Detectar Background i Foreground**

Detecció del que correspon al fons de la imatge (Background) i el que correspon als objectes (Foreground).

- **Detectar moviment dels objectes**

És conseqüència del requisit anterior. La part que correspon a Foreground seran objectes en moviment.

- **Detectar posició dels objectes**

És necessari pel sistema saber la posició dels objectes en tot moment.

- **Detectar l'aparició i la desaparició dels objectes**

El sistema ha de detectar si d'un frame²⁴ a l'altre ha aparegut o desaparegut un objecte de l'escena.

- **Diferenciar objectes de l'escena**

S'ha de discriminar els objectes diferents que hi ha dins l'escena. Per exemple, si detecta moviment de dos pilotes, ha de quedar clar que són dos objectes diferents.

- **Seguiment de trajectòries**

El sistema ha de fer un seguiment de les trajectòries de cadascun dels agents en moviment de l'escena.

- **Mostrar resultats**

Tot el tractament que fa sobre les imatges serà mostrat de forma visual per l'usuari.

4.4.2 Requeriments no funcionals

"Són restriccions imposades pel client o pel mateix problema i afecten al disseny" [7].

4.4.2.1. Requeriments de rendiment

- **Requeriments estàtics**

1. L'aplicació funcionarà en un únic ordinador a la vegada i amb un únic usuari simultani.

- **Requeriments dinàmics**

1. Ha de tenir un temps de resposta raonable. El resultat s'ha de mostrar al mateix temps que tracta cada frame independentment del tipus d'entrada (video/webcam)
2. Tolerància a errades i a accions incorrectes.

²⁴ Fotograma en anglès, és una imatge que pertany a una seqüència d'imatges.

3. Evitar en la mesura del possible falsos positius i falsos negatius.

4.4.2.2. Restriccions de disseny

- **Acompliment de normes**

1. Compliment de la Llei Orgànica 1/1982, de Protecció Civil del Dret a l'Honor, a la Intimitat Personal i Familiar i a la Pròpia Imatge. Es complirà de la següent forma: si es capten imatges des de la webcam, a l'acabar de processar-les una per una, s'eliminaran i no es guardaran enlloc. Tot i que, si s'utilitzen vídeos, aquests són descarregats de webs amb contingut legal d'Internet. Finalment, en el cas que apareguin persones serà sota el seu consentiment.

- **Restriccions hardware i software**

1. És imprescindible que l'ordinador que executi l'aplicació tingui suficients recursos per suportar un codi que tracta vídeo en temps real. Per tant, seria convenient un PC amb un mínim d'un gigabyte de memòria RAM aproximadament.
2. El sistema es desenvoluparà per funcionar sota S.O. Windows.
3. Tot el codi es farà en llenguatge de programació MATLAB (inclosa la interfície).
4. Ha de funcionar amb qualsevol webcam de tipus USB 2.0 (no es contemplarà la opció webcam IP) amb un mínim de resolució²⁵ de 640x480.

4.4.2.3. Interfícies externes

- **Requeriments sobre interfícies externes**

1. La interfície ha de ser fàcil d'utilitzar per a un usuari no expert. Per tant, ha de ser intuïtiva i senzilla.
2. Els resultats mostrats per la interfície han de ser clars i entenedors.

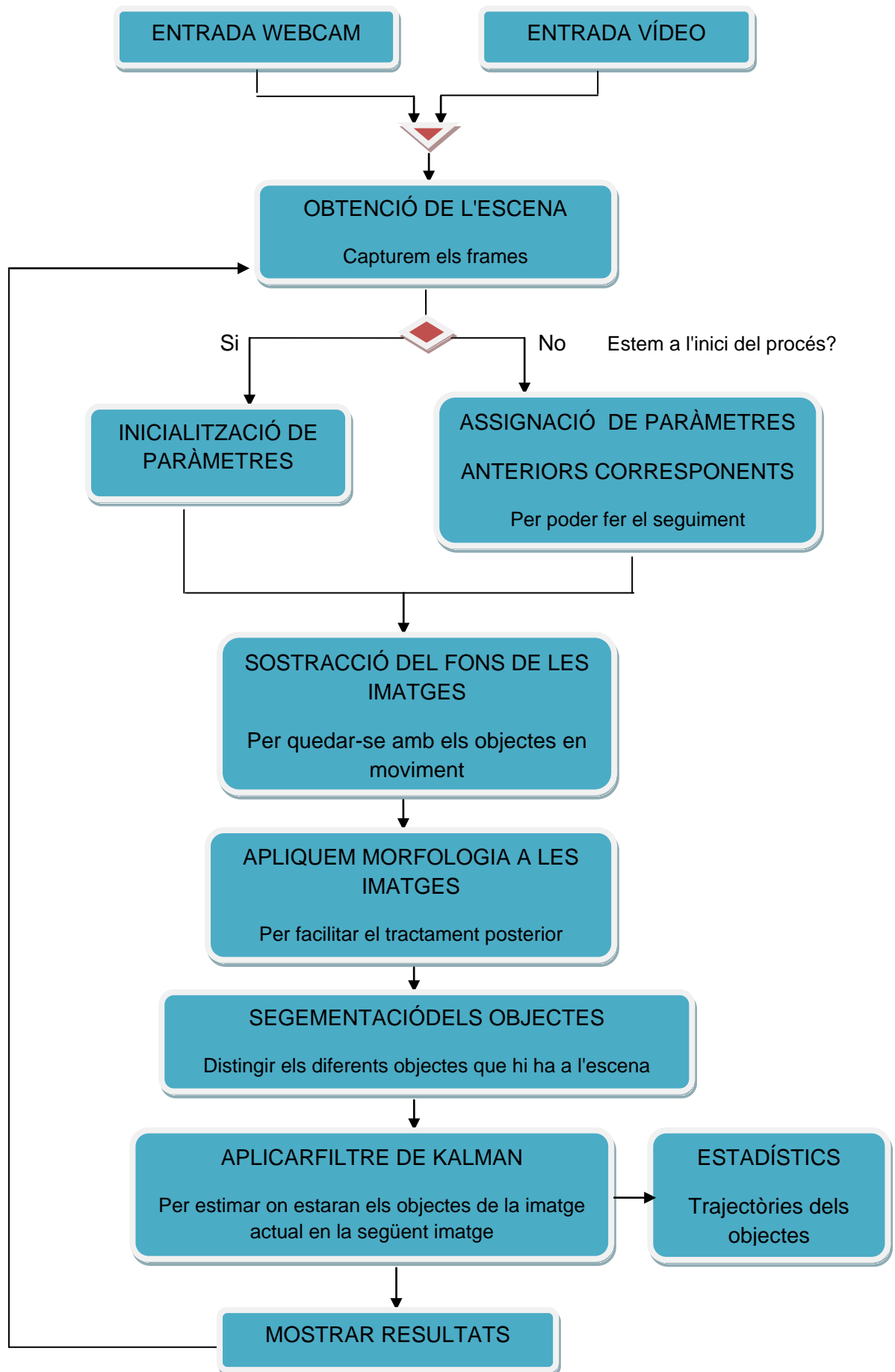
²⁵ Com més resolució, millors resultats s'obtidran.

4.5. Disseny modular de l'aplicació

A l'hora de trobar solució al problema plantejat s'ha escollit la programació modular com a paradigma de programació, és a dir, es parteix d'un problema gran amb una elevada complexitat i el que es fa és dividir-lo en problemes més petits que tinguin menys complexitat²⁶. Aquests s'anomenen mòduls.

Així per donar solució al problema del seguiment d'objectes segons una seqüència d'imatges d'entrada, s'ha dividit el codi en els següents mòduls (per fer-ho més comprensible s'ha fet un diagrama de flux que uneix tots els mòduls i mostra la relació entre ells):

²⁶ La famosa frase en el món de la programació: "Divideix i venç".



4.6. Disseny de la interfície

El disseny de la interfície està condicionat per l'especificació de requeriments, on s'ha parlat de les Interfícies externes i altres aspectes com els requeriments funcionals.

La interfície haurà de ser amigable i d'ús senzill per l'usuari no expert. A més, es valorarà més la funcionalitat que no pas el disseny gràfic. Altres aspectes a tenir en compte per satisfer requisits són:

- Haurà de tenir dos pantalles per mostrar els resultats: una per on mostrar el seguiment i l'altra per on mostrar les trajectòries. A més, els resultats han de ser intuïtius, és a dir, l'usuari ha d'entendre que està fent el programa (1,2 de l'esquema).
- Un menú per seleccionar el mode d'entrada webcam/vídeo (3 de l'esquema).
- Botons per poder modificar algun paràmetre que ens interessi si és adient (4 de l'esquema).
- Botons per iniciar i parar el procés (6 de l'esquema).
- En cas que el mode d'entrada sigui vídeo caldrà un menú per seleccionar quin vídeo volem tractar (5 de l'esquema).

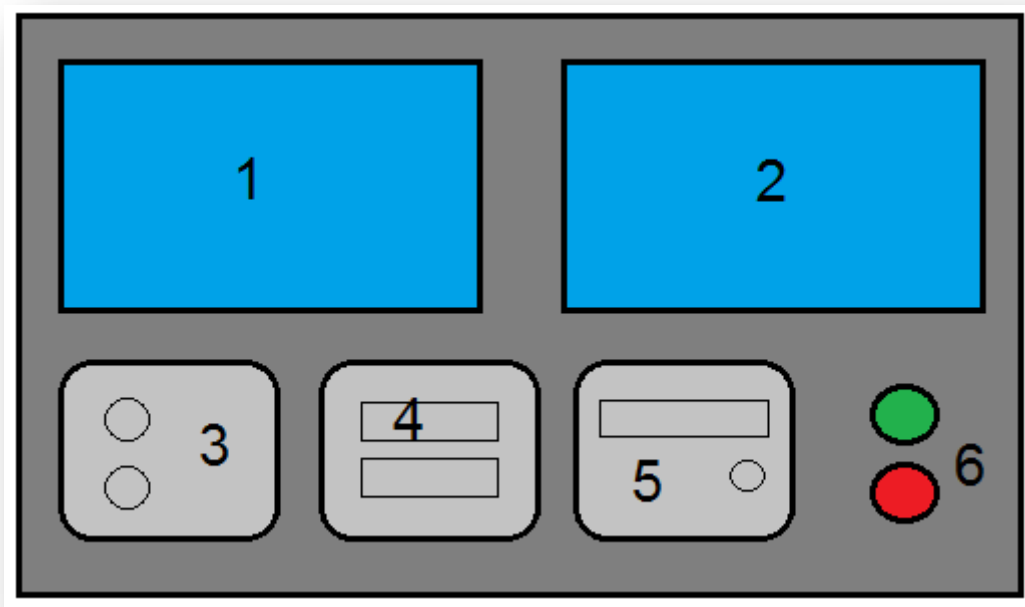


Figura 14: Disseny de la interfície.

CAPÍTOL 5

Desenvolupament del sistema

5.1. Introducció

En aquest punt bàsicament s'explicarà la implementació que s'ha dut a terme per assolir els objectius plantejats a partir del disseny. Per fer-ho, es realitzarà una descripció funcional dels mòduls principals. A més, si s'escau es comentaran les possibles traves que es poden trobar i com s'haurien de solucionar.

5.2. Descripció funcional

5.2.1. Background Subtraction

<i>Paràmetres d'entrada</i>	<i>Paràmetres de sortida</i>
<ul style="list-style-type: none">- Frame.- Threshold.- Velocitat.- Foreground²⁷ inicialitzat tot negre (imatge de ceros).	<ul style="list-style-type: none">- Foreground.

Aquest mòdul és la implementació de la part del disseny que correspon a *sostracció del fons de les imatges*. La seva explicació ja s'ha realitzat a l'apartat de fonaments teòrics.

Tot i així, cal mencionar que a la implementació duta a terme se li ha afegit un paràmetre anomenat "*Speed*", que controla amb quina velocitat s'actualitza el model de Background.

Aquí hi ha un exemple gràfic per entendre què fa la funció, on es veu el frame i el Foreground obtingut d'aquest:

²⁷ El Foreground creat serà una imatge binària, ja que és més senzill tractar amb imatges binàries que no pas amb imatges en escala de grisos.



Figura 15: Frame extret d'un dels vídeos per realitzar proves.



Figura 16: Foreground.

Com ja s'ha comentat, s'ha implementat el mètode *Approximate Median Method*. Del model implementat s'han de tenir en compte diversos aspectes:

- **Decidir el valor del paràmetre Threshold:** és un paràmetre el valor del qual no es pot saber a priori per a què sigui òptim, ja que tot depèn de la seqüència d'imatges concreta que s'estigui tractant (si té més o menys il·luminació, si els objectes es mouen molt o poc ràpid, si els objectes són més o menys grans, etc.). Per tant, normalment el que es fa és provar diversos valors i mirar amb quin s'obté millor resultat.

- **Decidir el valor del paràmetre Speed:** al principi s'utilitza un per defecte, però com passa amb el Threshold, depenent de la seqüència d'imatges serà més adient posar un valor o un altre.
- **Un objecte s'atura:** com s'ha vist anteriorment, és un dels casos de com pot evolucionar un objecte que s'està seguint. Si l'objecte que s'està seguint s'atura un temps prolongat, aquest passarà a formar part del model de Background i, per tant, quan es faci la extracció del fons l'objecte no serà part del Foreground perquè no el detectarà. També pot donar-se el cas que, si l'objecte s'atura un moment i de seguida continua movent-se, deixarà un rastre, que dependrà de la velocitat en què s'actualitza el model de Background.
- **Les ombres dels objectes:** si l'objecte en moviment està en una escena on l'ombra es veu molt, el mètode utilitzat no sabrà distingir què és l'ombra i no ho detectarà com a Background. No ha donat temps a implementar una solució per a aquest cas.

5.2.2. Morfologia

<i>Paràmetres d'entrada</i>	<i>Paràmetres de sortida</i>
- Foreground.	- Foreground preparat per a ser processat.

La funció de Background Subtraction retorna un Foreground amb brutícia, és a dir, queden restes que no interessin perquè no pertanyen a objectes en moviment. Es tracta, doncs, d'eliminar aquestes restes i deixar la imatge preparada pel processament següent.

Per fer-ho s'aplica la funció morfològica de l'erosió amb un element estructurant en forma de disc i s'obté el següent resultat:



Figura 17: Foreground després d'aplicar-li una erosió.

5.2.3. Distància mínima entre rectangles²⁸

<i>Paràmetres d'entrada</i>	<i>Paràmetres de sortida</i>
- Coordenades de les cantonades del rectangle A i B.	- Distància mínima entre els dos rectangles.

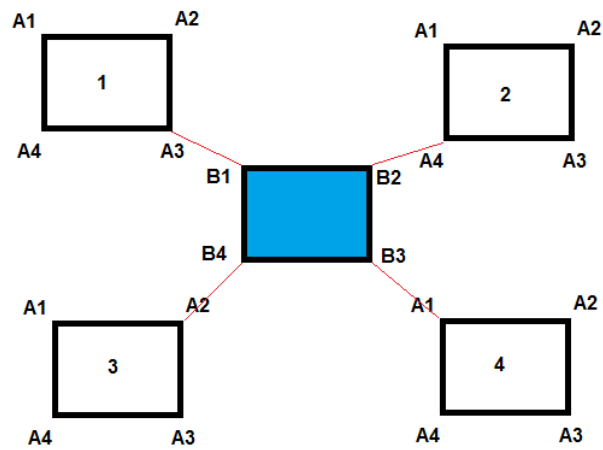
Com diu el seu nom aquesta funció calcula la distància²⁹ mínima que hi ha entre dos rectangles. Per fer-ho de forma senzilla, la explicació serà gràfica per mostrar els diferents casos que hi ha.

Cada cas mostra un rectangle *B*, de color blau, i les quatre possibilitats d'un altre rectangle *A*, de color blanc, que mostra, segons on estigui aquest, quina és la distància mínima respecte a *B*. Aquesta distància és indicada per la línia vermella que uneix els rectangles *A* amb el *B*.

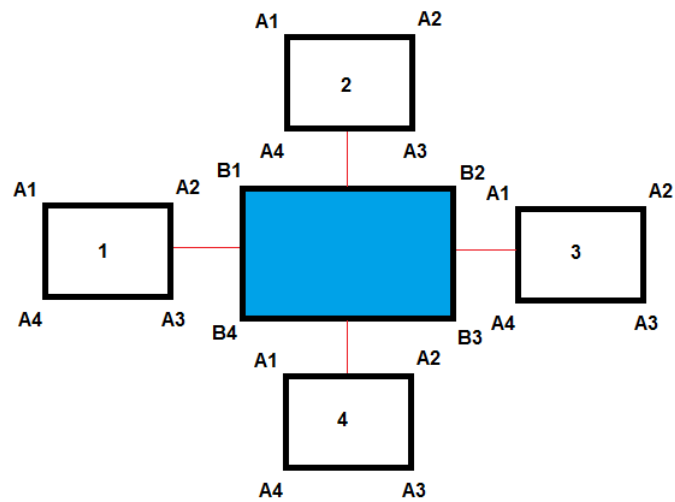
²⁸ Tot i semblar que és una funció que no té molta importància en el projecte s'ha volgut detallar el seu funcionament perquè és una funció que s'utilitza en dos mòduls claus per donar solució al problema plantejat.

²⁹ Nombre de píxels.

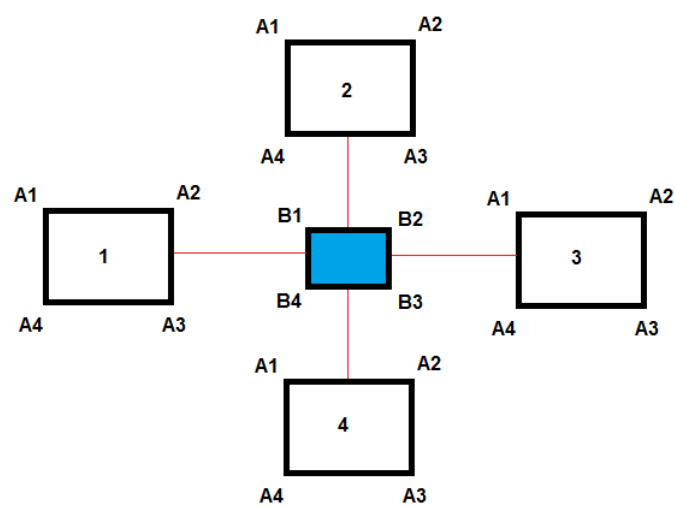
Cas 1:



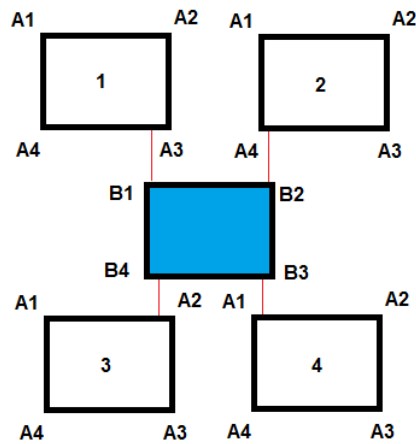
Cas 2:



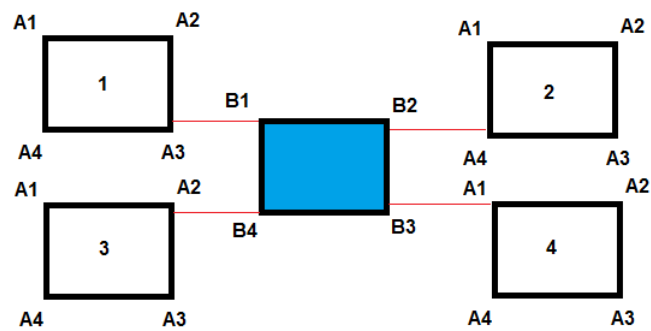
Cas 3:



Cas 4:

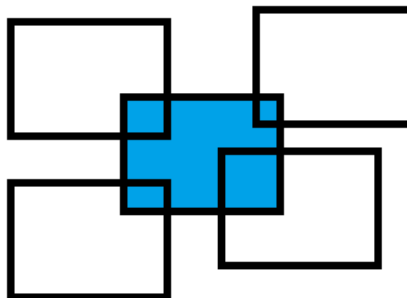


Cas 5:



Cas 6:

En qualsevol altre cas, voldrà dir que el rectangle B està inclòs o es creua amb el rectangle A i, en conseqüència, la distància entre els dos és zero.



5.2.4. Agrupar

<i>Paràmetres d'entrada</i>	<i>Paràmetres de sortida</i>
- Foreground.	- Rectangles mínims que enquadren els objectes del Foreground. - Coordenades dels centres d'aquests rectangles.

La funció *Agrupar* és molt important, ja que és el mòdul que s'encarrega de la part del disseny anomenada *segmentació dels objectes*. Fa la distinció dels diferents objectes de l'escena.

És una funció que, a partir de diversos components no connexes³⁰ d'un mateix objecte, els agrupa en un de sol. És a dir, la funció s'aplica al Foreground, la qual és una imatge binària on alguns objectes han quedat dividits en diversos trossos, no obstant, és necessari fer el seguiment d'aquestes parts com si d'un sol objecte es tractés, i d'això s'encarrega la funció, d'agrupar aquestes i retornar un rectangle que enquadra tot l'objecte i les coordenades del centre d'aquest.

Els passos que segueix la funció són:

- Primer s'agafa el Foreground i s'obtenen tots els Bounding Boxs de cada component no connex que hi hagi.

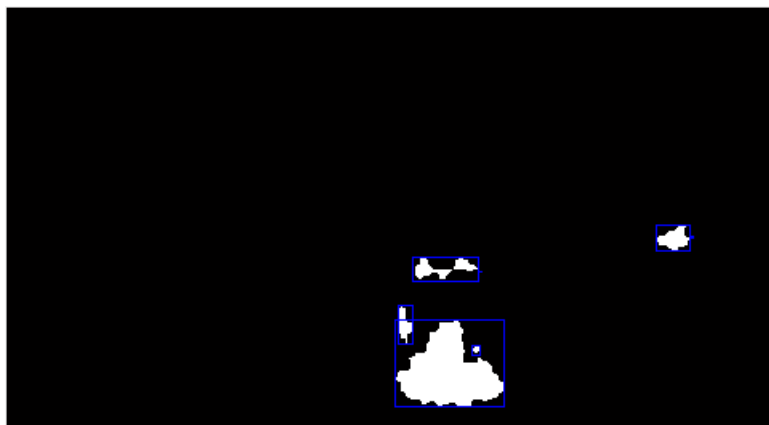


Figura 18: Foreground amb els Bounding Boxs.

³⁰ No estan connectats per cap píxel.

- Un cop s'ha trobat el rectangle mínim que enquadra cada component cal dir quins d'ells pertanyen a un mateix objecte, per fer-ho hi ha dos criteris diferents³¹:

1. *Segons la distància entre els rectangles*: es tracta de calcular la distància mínima entre tots els rectangles i després mirar si aquestes distàncies estan per sota d'un llindar màxim de distància que ens marcarà si dos components són d'un mateix objecte.

Per calcular la distància mínima entre tots els rectangles s'utilitza la funció explicada en el punt 5.2.3. i es crea una matriu amb totes les distàncies:

`matriuDistBetwRect =`

0	0	32.0000	0	39.0000	139.2049	170.1764
0	0	20.0250	50.0400	63.0714	207.9279	238.6629
32.0000	20.0250	0	54.0000	2.0000	148.0844	177.7217
0	50.0400	54.0000	0	61.0000	166.0030	196.7867
39.0000	63.0714	2.0000	61.0000	0	145.9932	176.2385
139.2049	207.9279	148.0844	166.0030	145.9932	0	2.0000
170.1764	238.6629	177.7217	196.7867	176.2385	2.0000	0

On per exemple, la fila 3, columna 2 indica que l'objecte 3 està a 20,05 píxels de distància de l'objecte 2. I la fila 1, columna 1, indica que l'objecte 1 està a distància 0 de l'objecte 1.

Ara es compara cada punt x,y de la matriu amb una distància llindar, per exemple 50, i cada punt que sigui inferior es posarà un 1 i en cas contrari un 0 quedant la matriu de la següent forma:

`matriuDistBetwRect =`

1	1	1	1	1	0	0
1	1	1	0	0	0	0
1	1	1	0	1	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
0	0	0	0	0	1	1
0	0	0	0	0	1	1

³¹ Tot i què els dos criteris estan implementats, només el primer s'aplica al projecte, l'altre es pot utilitzar però cal seguir les instruccions que diu al codi.

Es tracta d'una matriu simètrica de la qual s'han d'extraure els diferents conjunts que formen objectes.

Per trobar aquests conjunts s'ha de dividir la matriu en grups. Per a què dos files pertanyin al mateix grup han de tenir almenys un 1 en una mateixa columna. Per tant, en aquesta matriu hi ha dos grups. D'una banda hi ha el grup de les files (1,2,3,4,5) i, d'altra banda, el grup de les files (6,7). Cada grup conté quins objectes no connexes pertanyen al mateix objecte. Degut a això, ja només falta retornar un rectangle mínim que inclogui tots els Bounding Box dels objectes de cada grup i les coordenades del centre de cada rectangle, obtenint així el resultat final de la funció:

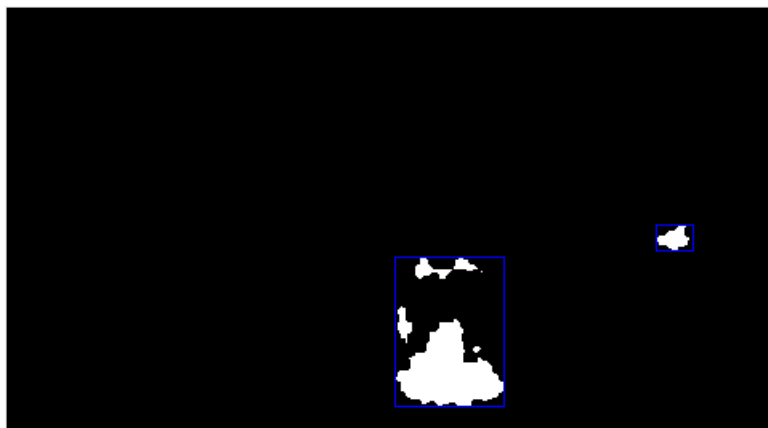


Figura 19: Foreground amb els rectangles que retorna la funció agrupar.

2. *Segons la distància entre els centres de massa dels objectes:* fa exactament el mateix que el criteri anterior, només canvia que les distàncies es calculen entre els centres de massa dels objectes i no s'utilitzen les distàncies mínimes entre els Bounding Boxes.

Aquesta funció té dos inconvenients:

- **Decidir el valor llindar de distància:** és impossible tenir un valor universal de llindar que ens determini si dos objectes en qualsevol escena (independentment de la mida dels objectes, si estan allunyats o no de la càmera, etc.) són dos components d'un mateix objecte. Per això, segons la

seqüència d'imatges que s'està tractant s'hauria de tenir en compte de posar un valor raonable.

Una possible solució que no ha donat temps a ser implementada és crear un mòdul que apliqui un procés de calibratge per trobar aquesta distància llindar segons el camp de vista³². És a dir, fent una relació entre la distància física real i la distància dels píxels.

- **La distància no sempre serà un bon criteri:** el fet de què dos objectes estiguin suficientment a prop, no implica obligatòriament que s'hagin de considerar el mateix objecte. Per exemple, si dues persones caminen separades, després s'apropen l'una a l'altra i es posen a caminar juntes, el mètode implementat ho detectarà com un sol objecte.

5.2.5. Kalman Filter

<i>Paràmetres d'entrada</i>	<i>Paràmetres de sortida</i>
<ul style="list-style-type: none"> -Estimador de la covariància "P". -L'estimador de l'estat x. -L'inicialitzador del filtre, indicarà si l'objecte és nou a l'escena o no. -Coordenades amb les mesures de la posició de l'objecte. 	<ul style="list-style-type: none"> -Estimador de la covariància "P" actualitzat. -L'estimador de l'estat x actualitzat. -L'inicialitzador de filtre actualitzat.

Com s'ha explicat a l'apartat de Fonaments Teòrics, el filtre de Kalman és capaç de pronosticar la posició futura d'un objecte en moviment a partir d'informació dels frames anteriors. Per exemple, si hi ha una pilota que es mou cap a la dreta quan s'aplica el filtre en temps "t" farà una estimació de què la pilota es troba encara més a la dreta:

³² El camp de vista és la quantitat d'escena que veiem a la imatge.



Figura 20: Exemple de funcionament del Filtre de Kalman.

La implementació de la funció ha sigut adaptada d'un codi ja existent³³ i està pensada per ser executada amb un sol objecte a la vegada. Al següent punt, es veurà com s'ha hagut d'aplicar a més d'un objecte per frame.

L'algorisme utilitzat és una simple traducció del pseudocodi vist als fonaments teòrics, és a dir, primer es fa la inicialització dels paràmetres necessaris per funcionar el filtre, després s'apliquen les equacions de predicció i correcció i finalment es retornen els paràmetres, els quals seran els paràmetres d'entrada de la següent crida a la funció.

Cal afegir que el codi té una variable anomenada "*kfini*" que és l'inicialitzador del filtre i controla si l'objecte que es vol tractar és (o no) nou a l'escena. Si no ho és, el filtre utilitza els paràmetres anteriors per inicialitzar l'estimador de l'estat x a priori, en cas contrari, li dona el valor de la mesura actual de la posició de l'objecte.

Durant la creació d'aquesta funció han aparegut diversos aspectes que s'han hagut de tractar:

- **Amb quin valor inicialitzar els paràmetres:** en primer lloc, es van mantenir tots els valors que venien per defecte en la funció, excepte el valor de la covariància de l'error mesurat " R ".

Aquest s'iniciava amb zeros i cada cop que es tractaven tots els objectes d'un frame es produïa una actualització del paràmetre de cara al següent frame. D'aquesta forma, es pretenia que el sistema en certa manera "aprengués".

³³ Exactament del codi utilitzat a l'enllaç [27].

Però, finalment, es va veure que el resultat no era idoni ja que les seqüències d'entrada amb què es van fer les proves, el moviment dels objectes tenien una durada de vida molt curta, és a dir, els vídeos tenien molt poca durada³⁴ i no donava temps a què el sistema calculés un bon valor. Per això, es va acabar utilitzant el valor per defecte de "*R*", ja que aquest estava calibrat i adaptat pel tipus de seguiment d'objectes que s'ha implementat en el projecte.

- **Filtre per més d'un objecte a la vegada:** amb això no es refereix a què simultàniament la funció s'apliqui a tots els objectes del frame en moviment, sinó que cada cop que s'apliqui a un objecte, s'ha de trobar la forma de poder mantenir els paràmetres d'aquest objecte per fer la crida al següent frame, ja que només funcionarà de manera correcta si se li assignen els paràmetres que li corresponen a aquest objecte i no els d'un altre. Per solucionar aquest problema es va implementar la funció *Compara Frames* explicada al següent punt.

5.2.6. Compara Frames

<i>Paràmetres d'entrada</i>	<i>Paràmetres de sortida</i>
<ul style="list-style-type: none"> -Frame actual. -Frame anterior. -Foreground actual. -Foreground anterior. -Rectangles actuals retornats per la funció Agrupar. -Rectangles anteriors retornats per la funció Agrupar. -Coordenades de la posició de cada objecte, predites en el frame anterior, pel filtre de Kalman. 	<ul style="list-style-type: none"> -Vector que indica per a cada objecte del frame actual, quin objecte era en el frame anterior. -Vector que indica si l'objecte no correspon a cap objecte del frame anterior, sinó que es tracta d'un objecte que ha aparegut nou a l'escena.

³⁴ La majoria no arribava al minut.

Es podria considerar el mòdul més important de tota la implementació del projecte, ja què gràcies a ell es pot fer el seguiment correcte de diversos objectes a la vegada.

Com s'ha vist al punt anterior, la funció del filtre de Kalman està pensada per només ser aplicada a un sol objecte, però en les seqüències d'imatges que s'han de tractar, a priori no se sap quants objectes hi haurà, per tant, la implementació ha de ser capaç de tractar tants objectes com hi hagi.

La funció consisteix en trobar una manera de mantenir els paràmetres que retorna el filtre de Kalman, cada cop que s'aplica a un objecte d'un frame al següent. Per exemple, si hi ha una seqüència d'imatges on hi apareixen dos objectes en moviment (*obj1* i *obj2*) i s'aplica el filtre de Kalman primer al "*obj1*" i després al "*obj2*", cadascun d'ells haurà de tenir el seus propis paràmetres d'entrada al filtre i quan es passi al frame següent i es torni a fer la crida, els paràmetres retornats per a la funció al frame anterior seran els paràmetres d'entrada de la funció en el frame actual. Per tant, l'entrada al filtre per "*obj1*" seran els retornats pel filtre aplicat a "*obj1*" abans. La següent figura ho explica gràficament:

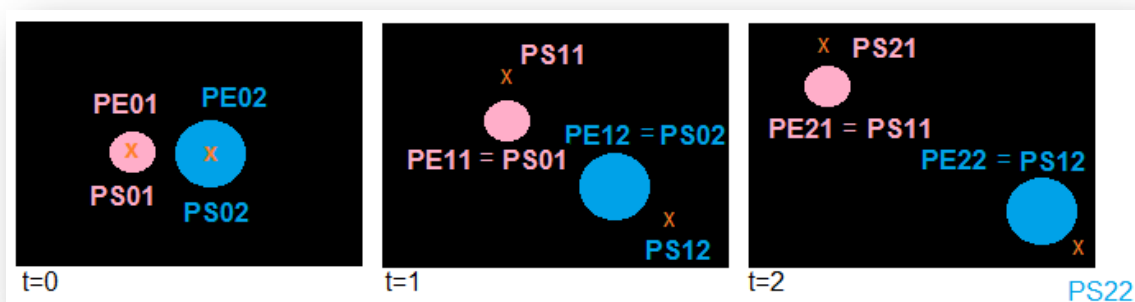


Figura 21: Correspondència entre paràmetres d'entrada i de sortida de Filtre de Kalman.

on la rodona rosa és l'objecte 1, la blava l'objecte 2, la "x" taronja és la posició futura que ha estimat el filtre, PE01 vol dir paràmetres d'entrada del frame en temps t=0 de l'objecte 1 i PS22 paràmetres de sortida del objecte 2 del frame en temps t=2.

Per tant, la funció *Compara Frames* retornarà a cada objecte del frame actual els paràmetres que li corresponguin, segons els objectes del frame anterior.

Llavors, es tractarà dels mateixos objectes i s'aplicarà de nou el filtre de Kalman.

Un cop s'ha entès què ha de fer la funció, és necessari explicar de quina forma ho aconsegueix. El criteri que s'ha seguit per assignar a un objecte del frame actual un del frame anterior ha sigut la **proximitat dels seus rectangles**³⁵ que l'enquadren. El procés que segueix l'algorisme és el següent (per facilitar l'explicació al lector es farà de la manera més gràfica possible):

1. S'agafa el Foreground anterior tractat per l'aplicació. Aquest té els rectangles que enquadren els objectes i la posició estimada del filtre de Kalman:

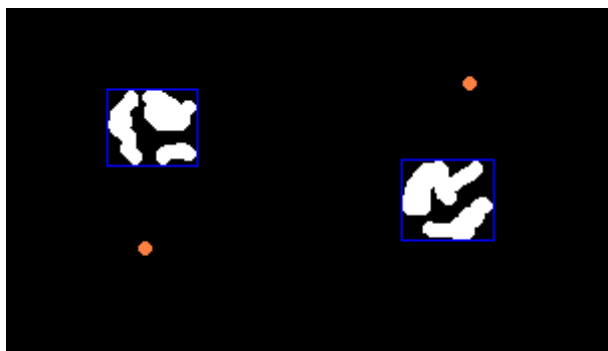


Figura 22: Foreground del frame anterior amb els objectes enquadrats i les estimacions del filtre de Kalman.

2. Es mou cada rectangle i es centra segons la posició estimada de cada objecte, quedant de la següent forma:

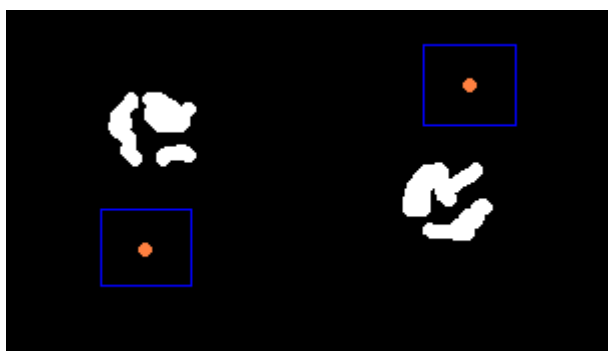


Figura 23: Foreground anterior amb els rectangles centrats en les estimacions.

³⁵ Recordem que aquests rectangles són els retornats per la funció Agrupar. Per tant, quan es parla d'objecte s'està referint a un conjunt de components no connexes que pertanyen al mateix objecte per criteri de proximitat.

3. Del Foreground actual només hi ha els objectes enquadrats³⁶:



Figura 24: Foreground actual amb els objectes enquadrats.

4. Es superposen els rectangles del Foreground actual amb els del Foreground anterior centrats en la posició estimada (per diferenciar-los, els del frame anterior són els de color groc):

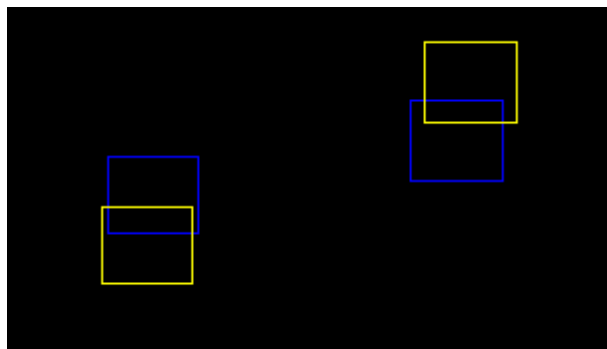


Figura 25: Rectangles del frame actual i de l'anterior superposats.

5. Finalment, s'ha de mirar de cada rectangle actual (blau) quin rectangle anterior (groc) té més a prop utilitzant la funció que calcula la distància mínima entre rectangles. Un cop calculat això, s'ha de comprovar que aquesta distància no superi una distància llindar, establerta arbitràriament. Si alguna d'aquestes distàncies supera el llindar vol dir que aquest objecte ha aparegut nou a l'escena. En aquest cas, no caldrà que rebi els paràmetres anteriors de cap objecte quan es faci la crida al filtre.
6. La funció retorna dos vectors: un, on cada posició diu quin objecte del frame anterior li correspon a cada objecte de l'actual i l'altre, que indica

³⁶ Ja que la crida a la funció Compara Frames és prèvia a la crida del filtre de Kalman.

de cada objecte del frame actual si ha aparegut nou a l'escena o ja estava en el frame anterior.

Barreres trobades i possibles solucions:

- **Escollir el valor llindar de la distància:** passa el mateix que a la funció Agrupar.

- **Escollir com a criteri la distància mínima:** no sempre la distància mínima assegura que l'objecte anterior sigui el mateix que l'actual.

Una possible solució per fer més robust el sistema i que ha sigut implementada a l'aplicació és: un cop obtingut el vector final retornat per la funció, on es diu que a cada objecte actual "X" li correspon un objecte anterior "Y" i que no és un objecte nou, es compara que, a més, tinguin semblants característiques relacionades amb el color. Hi ha dos criteris:

1. *Mirant com d'iguals o diferents són els píxels:* es tracta de calcular de cada objecte del frame actual i l'anterior la desviació estàndard³⁷ dels canals RGB, per així saber la variació de color que hi ha a cada canal. Un cop estan totes calculades, es compara cada objecte del frame anterior amb l'objecte que li correspon del frame actual. Si aquests dos tenen una variació de color molt diferent es diu que l'objecte "X" i "Y" no són el mateix objecte i que, per tant, "X" és un nou objecte aparegut a l'escena.
2. *Mirar la mitjana aritmètica:* és igual que el mètode anterior però en comptes de calcular la desviació, es calcula la mitjana aritmètica de cada canal i es compara amb la de l'altre objecte.

Aquests dos criteris són molt útils si els objectes que s'estan seguint tenen colors homogenis. Per exemple, si es segueix una pilota d'un sol color.

- **Un objecte es desfà en varis (canvis morfològics):** com s'ha dit anteriorment, ho sol causar la sostracció del fons. A més, la funció Agrupar interpreta que són diversos objectes³⁸. El que fa el mètode és tractar cada

³⁷ Es tracta de mesurar la variabilitat d'una variable aleatòria en torn a la seva mitjana.

³⁸ Ja que han superat la distància llindar.

objecte com a diferent i aplicar tot per igual, és a dir, buscarà quin rectangle té més a prop i retornarà quin objecte és. Per tant, normalment retornarà que cada tros ve del mateix objecte anterior.

- **Situacions d'oclusió:** dependrà del valor de la distància llindar de què quan torni a aparèixer l'objecte el detecti com l'anterior o com un de nou.

5.2.7. Procés

<i>Paràmetres d'entrada</i>	<i>Paràmetres de sortida</i>
-Una estructura anomenada "Proc", que prové de la interfície, que guardarà tota la informació necessària que es vol mantenir d'un frame al següent (només es manté informació del frame anterior). -Número de frame, per si l'entrada és un vídeo sàpiga quin frame ha de tractar cada cop que es crida la funció.	- L'estructura "Proc" actualitzada amb els canvis que s'han fet dins la funció en l'actual frame.

És el mòdul que s'encarrega de fer tot el processament de l'aplicació. Per dir-ho d'una altra forma, es tracta de fer les crides a les funcions en l'ordre que segueix el disseny modular de l'aplicació.

És a dir, primer detecta si el frame que ha de tractar prové d'un vídeo o de la webcam, crida la funció que sostrau el Background, crida a la funció que aplica morfologia, seguidament la que agrupa objectes i, després, a la que compara el frame actual amb l'anterior.

Aquest és un punt important, ja que la funció *Procés*, amb el resultat que retorna *Compara Frames*, s'encarrega d'actualitzar els paràmetres de cada objecte del frame actual, que seran l'entrada al filtre de Kalman, el qual també és cridat finalment en aquesta funció.

A part d'aquestes actualitzacions, la funció s'encarrega de mostrar els resultats que es van obtenint de forma intuïtiva:

➤ A una pantalla:

- Frame que s'està tractant.
- Els rectangles que enquadren els objectes. A més, fa que cada rectangle que pertanyi a un objecte diferent tingui un color diferent. D'aquesta forma, comptant els colors es pot saber el nombre d'objectes que hi ha a la seqüència.
- Una cadena, al costat de cada objecte, del tipus "*obj.num*" on "*num*" diu el número d'objecte que es tracta. Per no confondre l'usuari les lletres tenen el mateix color que el rectangle de l'objecte.
- Els resultats del filtre de Kalman. Un punt blau indica l'estimació de cada objecte.

➤ A l'altra pantalla:

- Es dibuixa la trajectòria de cada objecte. Sobre un fons negre es traça una línia que segueix la trajectòria que va fent cada objecte. Cada línia té el mateix color que el rectangle de l'objecte concret.

L'èxit de què la trajectòria es mostri correctament depèn directament de què la funció "Agrupar" retorni un bon resultat, ja que per fer la trajectòria es traça una línia que uneix la posició del centre de cada objecte d'un frame amb la posició del centre del mateix objecte al següent frame. Per tant, si la funció "Agrupar" un mateix objecte el retorna amb diferents rectangles, el dibuix de la trajectòria serà confús.

Degut a què és fa un tractament en temps real, s'ha modificat la forma en què es fan les crides que s'encarreguen de mostrar resultats (ja que aquestes gasten molts recursos). A més, si l'entrada prové de la webcam, abans de

mostrar les trajectòries hi ha un temps d'espera prudencial per esperar que la càmera s'adapti a l'escena.

Finalment, un cop fetes totes les crides adients i s'han mostrat els resultats, es guarden les variables que faran falta al frame posterior per continuar fent el seguiment dels objectes. Aquestes variables es guarden dins l'estructura "*Proc*".

CAPÍTOL 6

Conclusions

6.1. Introducció

Arribats a aquest capítol, és hora d'extreure les conclusions convenientes. La millor forma de fer-ho és comparant allò es va proposar dur a terme en un inici amb el que s'ha acabat realitzant al projecte.

En concret es mirarà si els objectius plantejats han estat assolits amb èxit o no. A més, s'analitzarà si s'ha seguit la planificació temporal inicial que subdividia les tasques del projecte o, pel contrari, hi ha hagut variacions. Seguidament, es parlarà sobre les possibles millores i ampliacions que es podrien aplicar al projecte i, per concloure, es farà una valoració personal global sobre el que ha significat pel projectista realitzar aquest treball.

6.2. Objectius

Els objectius plantejats inicials van ser:

- O.1.** Entendre i aprendre a utilitzar algunes de les tècniques de Visió per Computador relacionades amb el tractament de vídeo.
- O.2.** A partir de diverses seqüències d'imatges, amb diferents escenes, detectar on hi ha moviment i fer un seguiment dels agents d'aquesta escena.
 - O.2.1.** Seguiment de la trajectòria dels objectes en moviment.
- O.3.** Detectar quan apareix o desapareix un objecte a l'escena.
- O.4.** Distingir els objectes que hi ha dins l'escena.
- O.5.** Obtenir informació estadística del procés.

Ara s'analitzarà cada objectiu i es farà una reflexió sobre si s'ha assolit o no:

O.1. Es considera que s'ha assolit amb èxit. Abans de començar, el projectista tenia nocions bàsiques de Visió per Computador, però totes elles relacionades només amb imatges. En finalitzar, es pot dir que el projectista ha comprés el funcionament de tècniques de tractament de vídeo molt importants (com són el filtre de Kalman o la sostracció del fons de les imatges), a l'hora que ha après a utilitzar-les en el problema plantejat.

O.2. Aquest objectiu també s'ha realitzat exitosament. D'una banda, es volia detectar on hi ha hagut moviment dins la seqüència. D'això s'encarrega la funció que sostrau el fons a les imatges i ho fa eficientment. D'altra banda, es volia fer un seguiment dels objectes en moviment. S'ha aconseguit mitjançant la combinació de diverses funcions, en concret, el filtre Kalman i la funció "Compara Frames" (ambdues explicades anteriorment).

No obstant, també hi ha hagut problemes dels quals no s'han trobat solució. Per exemple, quan dos objectes en moviment s'ajunten una distància molt petita, el mètode implementat considera que són un mateix objecte, la qual cosa és incorrecte.

O.2.1. S'han assolit els objectiu marcats. En el mòdul "Procés" s'ha implementat una forma per mostrar les trajectòries dels objectes. Tot i que, com s'ha dit, depèn molt de la funció "Agrupar".

O.3. Funciona correctament. Cada cop que es tracta un frame de la seqüència es detecta si han aparegut o desaparegut objectes (realment el que es comprova és si l'objecte està en moviment, a la que desapareix o s'atura, es deixa de tractar aquest objecte).

O.4. Per distingir els objectes que hi ha dins l'escena s'ha implementat

la funció "Agrupar", que fa la seva funció correctament, sempre i quan es tinguin en compte les consideracions adients descrites a l'apartat "Descripció funcional".

O.5. Amb el dibuix de les trajectòries es mostra la recopilació de dades sobre el comportament dels objectes.

6.3. Planificació temporal final

La planificació temporal inicial feta a l'Estudi de Viabilitat no ha concordat massa amb la realitat. El diagrama de Gantt següent resumeix a grans trets com s'han dut a terme realment les tasques del projecte, mostrant només les que més temps han portat. (Nota: no apareixen dates, perquè tot i haver sigut regular a l'hora de fer la feina [12 hores setmanals] no es van controlar exactament les dates de finalització de cada tasca).

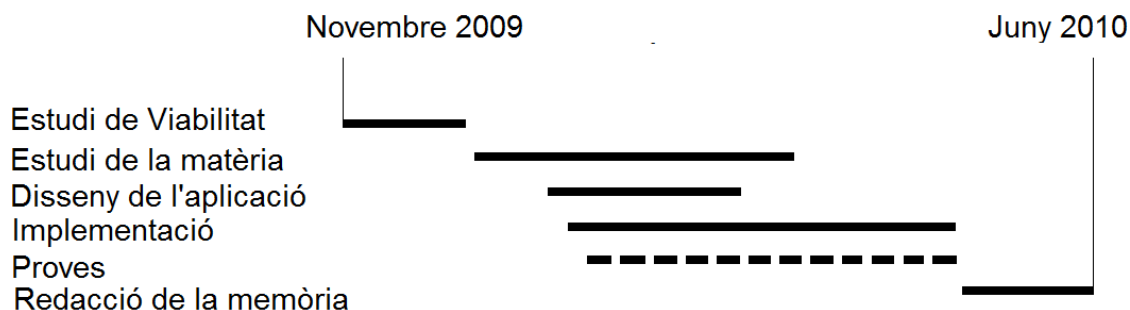


Figura 26: Diagrama de Gantt final.

Aquest diagrama pretén contrastar que a l'inici del projecte es va fer una planificació seqüencial, és a dir, una tasca nova es començava a l'acabar l'anterior però, realment, la realització de les tasques ha sigut com mostra l'esquema. Va caldre dedicar moltes hores a l'estudi de la matèria i, conforme s'anaven adquirint coneixements nous, s'anava creant el disseny de l'aplicació de forma evolutiva. Això vol dir que es dissenyava una part, seguidament s'implementava i es feien les proves corresponents. Llavors, segons els problemes que es trobaven es dissenyava una alternativa, es tornava a implementar i a provar fins que funcionés. I, finalment, un cop acabat el codi i

passades les proves, es va redactar aquesta memòria.

Per últim, afegir que la durada total del projecte s'ha allargat gairebé un mes, però s'ha acabat dins de la data límit d'entrega.

6.4. Futures ampliacions i millores

Es considera que aquest projecte té molta continuïtat futura. És a dir, un altre alumne podria continuar partint del treball realitzat en aquest projecte.

D'una banda, es podrien millorar aspectes que tenen una mica de feblesa. Creant nous mòduls que facin més robust i refinat el sistema, per així obtenir millors resultats. Un exemple on aplicar això podria ser a l'hora de mostrar les trajectòries, trobant un mètode que centri millor els objectes.

També es podrien substituir els mètodes utilitzats en el seguiment o en la sostracció del fons per altres més eficients, que en aquest projecte no s'han arribat a estudiar.

D'altra banda, també caldria tenir en compte el fet de realitzar millores de mètode més complexes com pot ser afegir nous models dinàmics perquè el sistema sigui més robust. Per exemple, que el sistema segueixi correctament moviments d'objectes amb rebots.

Finalment, es podria transformar el projecte actual, que està enfocat en un àmbit acadèmic ja que s'ha implementat un prototip, en una aplicació real. Per fer això caldria transformar tot el codi a un llenguatge de programació que sigui viable en aplicacions reals i no només en l'àmbit de laboratori, per exemple passar-ho a C++. A més a més, caldria millorar la interfície gràfica.

Per concloure, seria interessant a la vegada que força ambiciós, partint d'aquest projecte com a base, dissenyar un sistema de videovigilància intel·ligent, és a dir, centrar-se en el seguiment de persones i estudiar els seus moviments. Per això s'hauria de deixar de banda el tractament de vídeos guardats, que és en el que s'ha centrat el màxim d'esforç en el projecte, i caldria millorar el tractament si l'entrada prové en temps real des d'una

webcam.

6.5. Valoració personal

Per acabar el projecte m'agradaria fer una petita reflexió personal, així que parlaré en primera persona.

Al principi de curs va ser molt dur, ja que jo tenia clar que volia fer un projecte relacionat amb la Visió per Computador, però no hi havia cap proposta relacionada amb aquest tema. Per tant, vaig haver de buscar un professor que em volgués ajudar i a qui poder proposar un tema.

Un cop aconseguit això, van caldre moltes hores de dedicació diàries, ja que la base que tenia del tema del projecte era molt minsa. Gràcies a aquest esforç, considero que he madurat com a estudiant, he après molt i estic molt satisfet amb el treball realitzat tot i que el resultat no hagi estat òptim.

Per concloure, només em resta afegir que haver realitzat aquest projecte m'ha ajudat a decidir que vull continuar estudiant Visió per Computador i que la meva carrera professional s'encamini cap aquesta branca de la Informàtica.

Referències

- **Fonts electròniques**

[1] AliReza KashaniPour (2007, Març 13) “2D Target tracking using Kalman Filter” [Online] Diponible en <http://www.mathworks.com/MATLABcentral/fileexchange/14243-2d-target-tracking-using-kalman-filter> (2009, Decembre 4).

[2] Anònim, “Paradigmas de la Ingeniería de Software” [Online] Disponible en <http://www.ing.unp.edu.ar/asignaturas/estructuras/downloads/apuntes/clase23.pps> (2010, Maig 20).

[3] Anònim, “Análisis de los requerimientos de información” [Online] Disponible en http://148.202.148.5/cursos/cc321/fundamentos/unidad3/tema3_3_1.html (2010, Maig 20).

[4] Erick Jhonattan Rojas Figueroa, “Análisis de requerimientos” [Online] Disponible en <http://www.slideshare.net/edays/analisis-de-requerimientos-erick-rojas-figueroa> (2010, Maig 20).

[5] Dept. Ciències de la Computació de la Universitat Autònoma de Barcelona, “Enginyeria del Software I” [Online] Disponible en <http://www.cvc.uab.es/shared/teach/a20363/c20363.htm> (2010, Maig 20).

[6] Dept. Ciències de la Computació de la Universitat Autònoma de Barcelona, “Principis de l'enginyeria del software” [Online] Disponible en <http://www.cvc.uab.es/shared/teach/a20363/introES.pdf> (2010, Maig 20).

[7] Dept. Ciències de la Computació de la Universitat Autònoma de Barcelona, “Anàlisi de requeriments del software” [Online] Disponible en <http://www.cvc.uab.es/shared/teach/a20363/introAN.pdf> (2010, Maig 20).

[8] Anònim (2010, Juny 17) “Kalman Filter” [Online] Disponible en http://en.wikipedia.org/wiki/Kalman_filter#The_Kalman_filter (2010, Maig 26).

- [9] Álvaro Solera Ramírez (2003, Juliol) “*El filtro de Kalman*” [Online] Disponible en <http://www.bccr.fi.cr/ndie/Documentos/DIE-02-2003-NT-FILTRO%20DE%20KALMAN.pdf> (2010, Maig 26).
- [10] Anònim, “*Filtro Kalman-Bucy*” [Online] Disponible en http://iaci.unq.edu.ar/Materias/Cont.Digital/Apuntes/ApuntePagina/20-Filtro_de_Kalman.pdf (2010, Maig 26).
- [11] Greg Welch (2010, Abril 29) “*The Kalman Filter*” [Online] Disponible en <http://www.cs.unc.edu/~welch/kalman/index.html#Anchor-Rudolph-6296> (2010, Maig 26).
- [12] Doctorado en Tecnologías de las Comunicaciones (2003) “*El filtro de Kalman*” [Online] Disponible en <http://www.tsc.uc3m.es/~mlazaro/FiltAdapt/Kalman.pdf> (2010, Maig 26).
- [13] Patricia Rodríguez Muñoz (Juny 2003) “*Aplicación del filtro de Kalman al seguimiento de objetos en secuencias de imagenes*” [Online] Disponible en <http://www.escet.urjc.es/~jipantrigo/PFCs/MemoriaKalmanJun03.pdf> (2009, Novembre 22).
- [14] Greg Welch, Gary Bishop (2006, Juliol 24) “*An Introduction to the Kalman Filter*” [Online] Disponible en http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf (2010, Maig 26).
- [15] José Gregorio Díaz, Ana María Mejías. Francisco Arteaga, “*Aplicación de los filtros de Kalman a Sistemas de Control*” [Online] Disponible en <http://servicio.cid.uc.edu.ve/ingenieria/revista/a8n1/8-1-4.pdf> (2010, Maig 26).
- [16] Luis Baumela, “*Visión por Computador*” [Online] Disponible en <http://www.dia.fi.upm.es/~lbaumela/doctorado/> (2010, Maig 26).
- [17] Domingo Mery (2004, Agost 17) “*Visión por Computador*” [Online] Disponible en <http://dccplone.ing.puc.cl/dmery/dmery/repositorio/libros/ApuntesVisionComputador.pdf> (2010, Maig 26).

[18] Oscar Maldonado Valles (2002, Maig 13) “*Visión por Computadora*” [Online] Disponible en <http://www.depi.itch.edu.mx/apacheco/expo/html/ai11/vision.html> (2010, Maig 26).

[19] Universidad Rey Juan Carlos, “*Visión por Computador*” [Online] Disponible en <http://www.escet.urjc.es/~visionc/VisionPorComputador.pdf> (2010, Maig 26).

[20] CVC de la Universitat Autònoma de Barcelona, “*Sistemes de Visió Artificial*” [Online] Disponible en http://www.cvc.uab.es/shared/teach/a21308/apunts/Sistemes_de_visio.pdf (2010, Maig 26).

[21] Seth Benton (2008, Agost 10) “*Approximate Median*” [Online] Disponible en https://i.cmpnet.com/dspdesignline/2008/08/approximate_median.m (2010, Juny 5).

[22] Seth Benton (2008, Agost 10) “*Background subtraction, part 1: MATLAB models*” [Online] Disponible en <http://www.dspdesignline.com/210000460?printableArticle=true> (2010, Juny 5).

[23] CVC de la Universitat Autònoma de Barcelona, “*Imatges Binàries*” [Online] Disponible en <http://www.cvc.uab.es/shared/teach/a21308/apunts/ImatgesBinaries.pdf> (2010, Juny 6).

[24] CVC de la Universitat Autònoma de Barcelona, “*Morfologia Binària*” [Online] Disponible en http://www.cvc.uab.es/shared/teach/a21308/apunts/Morfologia_Binaria.pdf (2010, Juny 6).

[25] Noticias Jurídicas, “*Base de Datos de Legislación*” [Online] Disponible en http://noticias.juridicas.com/base_datos/Admin/lo1-1982.html#a1 (2010, Juny 16).

[26] Ariel Amato “*Background Subtraction Technique based on Chromaticity and Intensity Patterns*” [Online] Disponible en <http://figment.csee.usf.edu/~sfefilat/data/papers/TuBCT8.37.pdf> (2009, Novembre 12).

[27] José Francisco Vélez Serrano, Ana Belén Moreno Díaz, Angel Sánchez Calle, José Luis Esteban Sánchez-Marín, “*Visión por computador*” [Online] Disponible en <http://www.terra.es/personal/jfvelez/libro2/libro.html> (2009, Octubre 24).

[28] Robert Fisher (2007, Octubre 25) “*CAVIAR Test Case Scenarios*” [Online] Disponible en <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/> (2010, Juny 6).

- **Bibliografia**

[29] A. D. Marshall and R. R. Martin, “*COMPUTER VISION, MODELS and INSPECTION*”, World Scientific Series in Robotics and Automated Systems-Vol.4, 1993.

[30] J. Serra, “*IMATGE ANALYSIS AND MATHEMATICAL MORPHOLOGY Vol.1*”, Academic Press, 1993.

[31] Jordi Vitrià, “*Visió per computador*”, Universitat Autònoma de Barcelona Servei de Publicacions, 1995.

Agraïments

Al director del projecte, per acceptar la meva proposta. També per ajudar-me a "desencallar-me" a cada tutoria i guiar-me per poder avançar.

A la Berta, per tot el suport moral que m'ha brindat quan més frustrat estava i per ajudar-me en els dubtes i en la correcció de la memòria.

A l'Albert, per ajudar-me a gravar alguns dels vídeos que m'han servit per a les proves del projecte.

A la meva àvia Dora, per tots els matins que m'ha hagut de suportar en moments de desesperació mentre codificava el projecte i "m'encallava".

Finalment, als companys de classe, que sempre han estat allà per ajudar-me quan ho he necessitat.

David Perálvarez Fernández,

Sabadell, Juny de 2010.